



Il Metodo COSMIC di Misurazione della Dimensione Funzionale

Versione 3.0

Introduzione al Metodo

Settembre 2007

RINGRAZIAMENTI

Autori del Metodo "COSMIC-FFP"¹ Versione 2.0 (ordine alfabetico)

Alain Abran, École de technologie supérieure – Université du Québec,
Jean-Marc Desharnais, Software Engineering Laboratory in Applied Metrics - SELAM,
Serge Oigny, Bell Canada,
Denis St-Pierre, DSA Consulting,
Charles Symons, Software Measurement Services, UK

Revisori della versione 2.0, 1998/1999 (ordine alfabetico)

Moritsugu Araki, JECS Systems Research, Japan	Thomas Fecc.ke, Germany	Patrice Nolin, Hydro Québec, Canada
Fred Bootsma, Nortel, Canada	Eric Foltin, University of Magdeburg, Germany	Marie O'Neill, Software Management Methods, Ireland
Denis Bourdeau, Bell Canada, Canada	Anna Franco, CRSSM, Canada	Jolijn Onvlee, The Netherlands *
Pierre Bourque, , École de Technologie supérieure, Canada	Paul Goodman, Software Measurement Services, United Kingdom	Laura Primera, UQAM, Canada
Gunter Guerhen, Bürhen & Partner, Germany	Nihal Kececi, University of Maryland, United States	Paul Radford, Charismatek, Australia
Sylvain Clermont, Hydro Québec, Canada	Robyn Lawrie, Australia	Eberhard Rudolph, Germany
David Déry, CGI, Canada	Ghislain Lévesque, UQAM, Canada	Grant Rule, Software Measurement Services, United Kingdom*
Gilles Desoblins, France	Roberto Meli, Data Processing Organization, Italy	Richard Stutzke, Science Applications Int'l Corporation, United States
Martin D'Souza, Total Metrics, Australia	Pam Morris, Total Metrics, Australia*	Ilonar Sylva, UQAM, Canada
Reiner Dumke, University of Magdeburg, Germany	Risto Nevalainen, Software Technology Transfer Finland, Finland *	Vinh T. Ho, UQAM, Vietnam
Peter Fagg, United Kingdom	Jin Ng, Hmaster, Australia	

* Membri fondatori del COSMIC Core Team, insieme agli autori del metodo COSMIC-FFP.

¹ La versione 2.0 è stata la prima versione pubblicamente disponibile del metodo COSMIC-FFP, come chiamato in origine.

Revisori della versione 3.0, 2006/2007 (ordine alfabetico)		
Alain Abran, École de Technologie Supérieure, Université du Québec, Canada	Jean-Marc Desharnais, Software Engineering Lab in Applied Metrics – SELAM, Canada	Arlan Lesterhuis*, Sogeti, The Netherlands
Bernard Londeix, Telmaco, United Kingdom	Roberto Meli, Data Processing Organization, Italy	Pam Morris, Total Metrics, Australia
Serge Oigny, Bell Canada	Marie O'Neill, Software Management Methods, Ireland	Tony Rollo, Software Measurement Services, United Kingdom
Grant Rule, Software Measurement Services, United Kingdom	Luca Santillo, Agile Metrics, Italy	Charles Symons*, United Kingdom
Hannu Toivonen, Nokia Siemens Networks, Finland	Frank Vogelesang, Sogeti, The Netherlands	

* Curatori della versione 3.0 del metodo COSMIC.

Traduttori e revisori della versione 3.0 in italiano, 2011 (ordine alfabetico)		
Luigi Buglione Engineering.IT	Bruno Codan * SIDI	Marcello Dorian Donzella Edinform
Gianfranco Lanza * CSI Piemonte	Raffaele Russo Unisys	Andrea Salvatori NECS
Luca Santillo * Agile Metrics		

* Traduttori della versione italiana.

Copyright 2007. Tutti i diritti riservati. Il 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.

Una versione di pubblico dominio del Manuale di Misurazione COSMIC e altri documenti tecnici, incluse le traduzioni in altre lingue, sono disponibili presso il sito web www.cosmicon.com.

CONTROLLO DI VERSIONE

La seguente tabella riporta la storia delle versioni di questo documento.

DATA	REVISORE/I	Modifiche / Aggiunte
Settembre 2007	COSMIC Measurement Practices Committee	Prima versione disponibile pubblicamente. I contenuti sono parzialmente basati sul Capitolo 2 del Manuale di Misurazione, ma ampiamente rivisto ed esteso

PREFAZIONE

Il metodo COSMIC è un metodo standard per la misurazione della dimensione funzionale del software appartenente ai domini funzionali comunemente denotati come software “applicativo aziendale” (“business application”, o “gestionale” – “Management Information System”, MIS) e software “real-time”, e combinazioni dei due.

Il metodo COSMIC è stato accettato quale Standard Internazionale ISO/IEC 19761 “Software Engineering – COSMIC-FFP – A functional size measurement method” (in seguito indicato come “ISO/IEC 19761”) nel Dicembre 2002 dal Joint Technical Committee 1 / Sub-Committee 7 dell’ ISO/IEC (JTC1 / SC7).

Scopo del documento

Scopo di questo documento è fornire una panoramica del metodo di misurazione della dimensione funzionale COSMIC, versione 3.0. Il documento di “Introduzione al Metodo” è stato pensato per i lettori che:

- necessitano di una panoramica della metodologia, ma non hanno bisogno di conoscerne tutti i dettagli,
- non hanno esperienza nel campo della misurazione funzionale del software e hanno bisogno di un’introduzione sull’argomento,
- hanno familiarità con i metodi di misura funzionale di “prima generazione” (come i metodi “IFPUG”, “MkII” o “NESMA”) e stanno prendendo in considerazione l’adozione del metodo COSMIC.

Documentazione di riferimento

Per una lista completa della documentazione relativa al metodo COSMIC si faccia riferimento al documento “Documentation Overview and Glossary of Terms”. Il glossario di questo documento contiene le definizioni di tutti i termini comunemente utilizzati nei documenti COSMIC. Questo documento, come tutti gli altri documenti citati più sotto, possono essere scaricati gratuitamente all’indirizzo www.cosmicon.com.

Ulteriori documenti significativi a cui potrebbero essere interessati i lettori a cui è rivolto il presente documento sono i seguenti:

- lo standard ISO/IEC 19761, che contiene le definizioni e le regole normative fondamentali del metodo.
- il “Manuale di Misurazione COSMIC, versione 3.0”, che contiene le regole e le definizioni, fornisce ulteriori spiegazioni e una maggiore quantità di esempi, con lo scopo di aiutare il misuratore a comprendere ed applicare pienamente il metodo. Questo dovrebbe essere il principale “documento di lavoro” di cui il misuratore ha bisogno nella pratica operativa.
- l’“Advanced and Related Topics”, che contiene documentazione di livello più avanzato, relativa alle stime sulla dimensione funzionale effettuate nelle prime fasi del ciclo di vita di un progetto e sulla conversione secondo il metodo COSMIC di misure funzionali effettuate con altri metodi

Il resto della documentazione COSMIC disponibile include le Linee Guida per l’applicazione del metodo in specifici domini software, vari casi di studio che illustrano l’utilizzo del metodo, articoli di ricerca, dati di benchmark, ecc. Sono disponibili anche traduzioni in altre lingue del Manuale di Misurazione. Tutti questi documenti sono disponibili all’indirizzo www.cosmicon.com.

Informazioni più generali sulle misure funzionali e sui suoi utilizzi, sui vantaggi del metodo COSMIC, sull’organizzazione COSMIC e sulle sue attività, sui fornitori di servizi relativi al metodo COSMIC, la newsletter COSMIC ecc., possono essere trovate all’indirizzo www.cosmicon.com.

Il COSMIC Measurement Practices Committee

INDICE

1	INTRODUZIONE	7
2	INTRODUZIONE AL METODO DI MISURAZIONE COSMIC.....	9
2.1	Applicabilità del metodo COSMIC.....	10
2.1.1	<i>Domini di applicabilità</i>	<i>10</i>
2.1.2	<i>Domini di non applicabilità</i>	<i>10</i>
2.2	I modelli software COSMIC.....	10
2.2.1	<i>Requisiti Utente Funzionali</i>	<i>10</i>
2.2.2	<i>Il Modello del Contesto del Software COSMIC.....</i>	<i>11</i>
2.2.3	<i>Il Modello Generale del Software COSMIC</i>	<i>18</i>
2.3	Panoramica del processo di misurazione COSMIC.....	21
2.3.1	<i>Fase di Strategia della Misurazione.....</i>	<i>21</i>
2.3.2	<i>Fase di Mappatura</i>	<i>23</i>
2.3.3	<i>Fase di Misurazione.....</i>	<i>23</i>
	APPENDICE A – PROCEDURA PER RICHIESTE DI MODIFICA E COMMENTI	25

INTRODUZIONE

Il software è una delle maggiori voci di molti budget aziendali. Le aziende riconoscono l'importanza di controllare le spese per il software e analizzare i risultati di quanto investito nello sviluppo e nella manutenzione dello stesso, per confrontarli con i riferimenti di mercato. Da qui nasce la necessità di analizzare sia la qualità che la produttività associate allo sviluppo e alla manutenzione del software.

Da un lato, le misurazioni tecniche sono necessarie agli sviluppatori per quantificare le performance tecniche dei prodotti o dei servizi. Le misure tecniche possono essere utilizzate, ad esempio, per analizzare l'efficienza o migliorare le performance di disegno, ecc.

Dall'altro, le misurazioni funzionali sono necessarie ad esempio agli sviluppatori per stimare o misurare la dimensione del software a partire dai requisiti, nelle prime fasi del ciclo di vita di un progetto, o come input principale per stimare l'impegno di un progetto, o ancora per quantificare le prestazioni di un prodotto o di un servizio dal punto di vista dell'utente in modo da poterne analizzare la produttività. Le misure funzionali devono essere indipendenti sia dallo sviluppo tecnico che dalle scelte implementative. Possono quindi essere utilizzate per confrontare la produttività di applicazioni realizzate per mezzo di differenti tecniche e tecnologie.

La Function Point Analysis (FPA)² è un possibile metodo di misurazione della dimensione funzionale. È utilizzabile nel dominio software delle "applicazioni aziendali" (MIS – Management Information Systems), dove è stata abbondantemente utilizzata in analisi e stime della produttività (Abran, 1996; Desharnais, 1988; Jones, 1996; Kemerer, 1987). Essa può catturare con successo le particolari caratteristiche funzionali del software MIS.

La FPA è stata tuttavia criticata per non essere universalmente applicabile a tutti i tipi di software [Conte, 1986; Galea, 1995; Grady, 1992; Hetzel, 1993; Ince, 1991; Jones, 1988; Jones, 1991; Kan, 1993; Whitmire, 1992]. In particolare, la FPA non è stata ben accolta nella comunità del software real-time.

Il metodo "Full Function Point" (versione 1.0) è stato proposto nel 1997³ con lo scopo di estendere la FPA per includere la dimensione funzionale del software real-time e di sistema. Test sul campo hanno dimostrato che il metodo FFP è adatto a misurare anche la dimensione funzionale del software MIS, portando a risultati simili alla FPA per applicazioni appartenenti a quel dominio⁴.

Nel 1998, il gruppo FFP è confluito nel gruppo COSMIC⁵ che proponeva i principi per una seconda generazione di metodi di misurazione della dimensione funzionale. Il risultato di questa attività è stata una prima versione di prova, disponibile al pubblico, del metodo di misurazione COSMIC – FFP, pubblicata nell'Ottobre 1999.

² Albrecht A.J., Gaffney Jr. J.E., "Software function, source lines of code and development effort prediction: a software science validation", IEEE Transactions on Software Engineering, Vol. SE-9, pp. 639-648, Novembre 1983. 'FPA' s now known as the 'FPUG' method.

³ St-Pierre D., Maya M., Abran A., Desharnais J.-M., Bourque P., "Full Function Points: Counting Practices Manual", Technical Report 1997-04, Université du Québec à Montréal, Montréal, Canada. Disponibile sul Web all'URL: www.lrgl.uqam.ca/ffp.html

⁴ Oligny, S.; Abran, A.; Desharnais, J.-M.; Morris, P., *Functional Size of Real-Time Software: Overview of Field Tests*, in Proceedings of the 13th International Forum on COCOMO and Software Cost Modeling, Los Angeles, CA, October 1998.

⁵ Si veda www.cosmicon.com per ulteriori informazioni.

Dalla versione 2.0 in avanti, il metodo di misurazione è stato pensato per essere pienamente aderente sia allo standard ISO/IEC 14143-1:1998 (e alla successiva versione ISO/IEC 14143-1:2007) che ai principi generali del COSMIC.

Dalla versione 3.0, il nome del metodo è stato semplificato da “COSMIC-FFP” a “COSMIC”,

L’iniziativa COSMIC

Data la crescita esplosiva e la varietà dei contratti e dell’outsourcing del software, sia i clienti che i fornitori necessitano di metodi più accurati per la stima e la misurazione delle performance. Questi metodi devono essere affidabili indipendentemente dal tipo di software esaminato. I metodi di misurazione di “prima generazione” non sempre hanno una robustezza sufficiente a soddisfare le esigenze del mercato, o funzionano solo per specifici tipi di software. L’industria ha urgente bisogno di misure funzionali del software che siano sia più accurate sia più usabili universalmente.

Il gruppo COSMIC ha lo scopo di soddisfare sia le richieste dei fornitori, che si occupano di tradurre i requisiti utente nella dimensione del software dal produrre come tappa fondamentale nel processo di stima dei costi, sia quelle dei clienti che considerano la dimensione funzionale del software acquistato un indice importante per valutare le performance del proprio fornitore.

Il COSMIC (COmmon Software Measurement International Consortium) è un’iniziativa volontaria di un gruppo internazionale di esperti nella misurazione del software, provenienti sia dal settore industriale che accademico, provenienti dall’Asia, dall’Europa e dal Nord America. Lo scopo originale del progetto COSMIC era quello di sviluppare, testare e proporre al mercato un nuovo metodo di misurazione funzionale per supportare sia la stima che la misura delle performance. Questo scopo è stato raggiunto e il metodo viene ora utilizzato a livello mondiale da un numero crescente di organizzazioni del settore sia pubblico che privato.

I principi del COSMIC sono stati definiti nel 1999 e sono stati seguiti da test sul campo, condotti con successo nel 2000-2001 sia da aziende di livello internazionale che da istituzioni accademiche. Articoli che illustrano i risultati di questi test e molti altri risultati della ricerca sono disponibili sul sito www.cosmicon.com. Il processo di definizione di uno standard internazionale per il metodo COSMIC è iniziato nel 2001. Lo standard è stato approvato nel Dicembre 2002 ed è stato pubblicato dall’ISO nei primi mesi del 2003 come ISO/IEC 19761.

COSMIC continua a migliorare le definizioni e le spiegazioni relative al metodo alla luce delle esperienze pratiche, tuttavia **si vuole evidenziare che il Modello Generale del Software, che sta alla base della misurazione, non è stato modificato dalla sua prima pubblicazione nel 1999**. La versione 3.0 del Manuale di Misurazione rappresenta l’ultima tappa in questo processo di miglioramento continuo ma sempre compatibile con lo standard ISO/IEC 19761. La designazione di “versione 3.0” confrontata con la precedente “versione 2.2” indica che la 3.0 rappresenta un passo importante nel miglioramento del metodo. Per una lista completa dei cambiamenti introdotti nel passaggio dalla versione 2.2 alla versione 3.0 si veda il “Manuale di Misurazione 3.0”.

Il Common Software Measurement International Consortium (COSMIC) prevede di sottoporre queste aggiunte e miglioramenti all’ISO per includerli nello standard ISO/IEC 19761 quando sarà sottoposto a revisione nel 2008.

Nel 2006, il COSMIC ha introdotto l’esame di certificazione “Entry-level” per i misuratori che usano il metodo. Coloro che usano il metodo COSMIC sono invitati a sottoporre i risultati delle misurazioni effettuate sui loro progetti al database dell’ International Software Benchmarking Standards Group (“ISBSG”), per ampliare i dati di benchmark relativi alle misurazioni effettuate utilizzando il metodo COSMIC.

Ulteriori informazioni su COSMIC, le sue pubblicazioni, le sue attività e le sessioni di esame sono disponibili sul sito www.cosmicon.com. Per ulteriori informazioni sull’ISBSG si veda il sito www.isbsg.org.

INTRODUZIONE AL METODO DI MISURAZIONE COSMIC

Il metodo COSMIC definisce uno standard per la misura della dimensione funzionale del software. Il presente capitolo descrive:

- il tipo di software per cui il metodo è stato ideato (conosciuto anche come il “dominio di applicabilità” del metodo) nel paragrafo 2.1
- i modelli software utilizzati nella misurazione, nel paragrafo 2.2. Questi modelli presentano tutti i concetti di base del metodo COSMIC. La comprensione di questi concetti è importante perché, per misurare la dimensione funzionale di una applicazione reale, il misuratore deve mappare i manufatti prodotti (ad esempio i suoi requisiti o la sua installazione) sui concetti dei modelli COSMIC
- il processo di misurazione COSMIC, che consiste di tre fasi:
 - la Strategia della Misurazione, effettuata prima di iniziare la misura vera e propria (paragrafo 2.3.1)
 - la Fase di Mappatura (paragrafo 2.3.2)
 - la Fase di Misurazione (paragrafo 2.3.3)

Il risultato del processo di misurazione è un valore espresso in “COSMIC Function Point” (o “CFP”). Le fasi sopra elencate sono illustrate in Fig. 2.0.



Figura 2.0 – Struttura del metodo COSMIC.

I paragrafi 2.3.1, 2.3.2 e 2.3.3 sono dettagliati ciascuno in uno specifico Capitolo del Manuale di Misurazione, dove vengono fornite definizioni complete e dettagliate, principi e regole del metodo, accompagnandole con esempi dimostrativi.

2.1 Applicabilità del metodo COSMIC

2.1.1 Domini di applicabilità

Il metodo COSMIC per la misurazione della dimensione funzionale è progettato per essere applicabile alle funzionalità software dei seguenti domini:

- software applicativo aziendale, tipicamente richiesto a supporto della gestione dei processi aziendali, come il software bancario, assicurativo, per la contabilità, il personale, gli acquisti, la distribuzione o la produzione. Questo tipo di software è spesso denotato come “a prevalenza di dati” (“data rich”), poiché la sua complessità è largamente dominata dalla necessità di gestire grandi quantità di dati riguardanti eventi nel mondo reale.
- software real-time, avente il compito di seguire o governare eventi che accadono nel mondo reale. Esempi sono il software per centraline telefoniche e per commutazione di messaggi, il software integrato (“embedded”) in periferiche di controllo di macchinari come elettrodomestici, ascensori, motori di automobili e velivoli, il software per controllo di processo e acquisizione automatica di dati e il software incorporato nei sistemi operativi per computer.
- combinazioni delle tipologie sopra elencate, come per es. nei sistemi di prenotazione in tempo reale per le linee aeree o gli alberghi.

2.1.2 Domini di non applicabilità

Il metodo di misurazione COSMIC non è ancora stato progettato per prendere in considerazione le funzionalità di software “mathematically-intensive”, software cioè caratterizzato da algoritmi matematici complessi o da altre regole specializzate e complesse, come quelle presenti nei sistemi esperti, nel software di simulazione, nel software ad auto-apprendimento, nei sistemi di previsioni meteorologiche, ecc., o che elaborano variabili continue quali suoni (audio) o immagini (video), come quelle contenute per es. nel software per videogiochi, negli strumenti musicali, ecc. Il metodo COSMIC non misura nemmeno aspetti come la “complessità” (comunque definita), che possa contribuire alla dimensione del software.

Per tali tipi di software è possibile, tuttavia, definire estensioni locali del metodo di misurazione COSMIC. Il Manuale di Misurazione illustra in quali contesti si dovrebbero usare tali estensioni locali e ne fornisce degli esempi.

2.2 I modelli software COSMIC

Questa sezione fornisce una panoramica del metodo COSMIC e dei suoi concetti di base. La definizione di tutti questi concetti viene data nel documento “Documentation Overview and Glossary of Terms”. In questa sezione, la prima volta che viene usato un termine legato a questi concetti, viene scritto in **grassetto**.

È essenziale che i misuratori comprendano TUTTI i modelli COSMIC e i concetti base descritti più sotto, e siano capaci di applicare TUTTI i principi e le regole descritte nei capitoli del Manuale di Misurazione quando effettuano una misurazione. Solo in questo modo il misuratore potrà essere sicuro che le misurazioni effettuate siano affidabili e ripetibili da altri misuratori e/o possano essere confrontate con misurazioni effettuate in altri ambienti software.

2.2.1 Requisiti Utente Funzionali

Il metodo di misurazione COSMIC comporta l'applicazione di un insieme di modelli, principi, regole e procedure ai **Requisiti Utente Funzionali (Functional User Requirements, FUR)** di una data porzione di software. Il risultato è un “valore (numerico) di una quantità” (come definito dall'ISO), che rappresenta la **dimensione funzionale** della porzione del software secondo il metodo COSMIC.

La dimensione funzionale misurata con il metodo COSMIC è progettata per essere indipendente da qualsiasi scelta implementativa racchiusa nei manufatti operativi del software da misurare. Le “funzionalità” si riferiscono alle “elaborazioni di informazioni che il software deve svolgere per i propri utenti”.

Più precisamente, un'affermazione dei FUR descrive “cosa” il software deve fare per gli **utenti funzionali** che sono i mittenti e i destinatari designati dei dati da e verso il software. Un'affermazione dei FUR esclude qualsiasi requisito tecnico o qualitativo che affermi “come” il software deve svolgersi. Quando si misura una dimensione funzionale, si prendono in considerazione solo i FUR.

Estrarre i requisiti utente funzionali dai manufatti del software nella pratica

Nel mondo reale dello sviluppo software è raro trovare dei manufatti per il software in cui tutti i FUR siano chiaramente distinti da altri tipi di requisiti e siano espressi in una forma adatta per una misurazione diretta senza alcuna necessità di interpretazione. Questo significa che abitualmente il misuratore dovrà estrarre i FUR così come sono forniti oppure presenti implicitamente negli effettivi manufatti del software, prima di mapparli sui concetti dei “modelli del software” del metodo COSMIC.

I Requisiti Utente Funzionali possono essere derivati dai documenti di ingegneria del software che sono realizzati ancor prima del software vero e proprio, come le “Specifiche dei Requisiti”, i risultati dell'analisi funzionale dei requisiti, ecc. Pertanto la dimensione funzionale del software può essere misurata prima della sua implementazione in un sistema informatico..

In qualche caso, il software dovrà essere misurato con poca, o nessuna, documentazione disponibile e i FUR potrebbero non essere documentati (ad esempio, per i software “legacy”). In questi casi è comunque possibile estrarre i FUR dal software anche se questo è già stato implementato e installato, ad esempio esaminando i report prodotti o il flusso dei dati.

Estrarre o derivare i requisiti utente funzionali dai manufatti del software

Il processo per estrarre i FUR dai diversi tipi di manufatti di ingegneria del software, oppure per dedurli dal software installato, e per esprimerli nella forma richiesta per misurarli con il metodo COSMIC, varierà ovviamente da caso a caso. Questi processi dipendono dal dominio software a cui appartengono e variano così tanto l'uno dall'altro che non possono essere tutti trattati dettagliatamente dal metodo COSMIC. Si assume che i requisiti utente funzionali del software oggetto di misurazione o esistano già o possano essere estratti o derivati dai relativi manufatti, in funzione dello scopo della misurazione. In ogni caso il COSMIC pubblica anche delle “Linee Guida”, che descrivono aspetti caratteristici dell'estrazione dei FUR per software legati a specifici domini.⁶

Il Manuale di Misurazione si limita quindi a descrivere e definire i concetti dei modelli del software COSMIC, cioè gli obiettivi dell'estrazione o deduzione³. Questi concetti sono incorporati in un insieme di principi esposti in due modelli del software COSMIC, il “Modello del Contesto del Software” e il “Modello Generale del Software”.

2.2.2 Il Modello del Contesto del Software COSMIC

Una porzione di software oggetto di misurazione secondo il metodo COSMIC deve essere ben definita (nell'**ambito** della misurazione) e tale definizione deve tener conto del suo contesto relativamente a qualsiasi altro software e/o hardware con cui essa interagisce. Il Modello del Contesto del Software introduce i principi e i concetti necessari per tale definizione.

⁶ Il documento “Guideline for Sizing Business Application Software using COSMIC” fornisce una guida alla mappatura di diversi metodi di analisi dei dati e individuazione dei requisiti utilizzati nel dominio degli applicativi aziendali sui concetti del metodo COSMIC

PRINCIPI – Il Modello del Contesto del Software COSMIC

- a) Il software è delimitato dall'hardware
- b) Il software è tipicamente strutturato in **strati**
- c) Uno strato può contenere una o più porzioni di software “alla pari” separate e ogni porzione di software può ulteriormente consistere di **componenti alla pari** separate
- d) Ogni porzione di software oggetto di misurazione è individuata dal proprio ambito di misurazione, che deve essere confinato interamente in un singolo strato
- e) L'ambito di una porzione di software oggetto di misurazione dipende dallo **scopo** della misurazione
- f) Gli utenti funzionali di una porzione di software sono identificati dai requisiti utente funzionali della porzione di software oggetto di misurazione come i mittenti e/o destinatari designati dei dati
- g) Una porzione di software interagisce con i propri utenti funzionali tramite **movimenti di dati** attraverso un **confine** e può muovere i dati da e verso la **memoria permanente** all'interno del confine
- h) I FUR del software possono essere espressi a differenti livelli di granularità
- i) Il livello di granularità al quale si dovrebbero effettuare normalmente le misurazioni è quello dei **processi funzionali** (vedi paragrafo 2.2.3)
- j) Se non è possibile misurare al livello di granularità dei processi funzionali, i FUR del software dovrebbero allora essere misurati con un approccio di approssimazione e convertiti al livello di granularità dei processi funzionali⁷

Questi principi e i relativi concetti saranno dettagliati e illustrati mediante esempi nel seguito. A questo scopo, si devono distinguere due punti di vista di un sistema hardware / software, cioè del contesto in cui una porzione di software viene misurata, ovvero:

- La struttura “fisica”, che mostra come il software sia tipicamente strutturato in una gerarchia di strati, ognuno con le sue funzioni particolari. Questa struttura mostra che, in realtà, ogni comunicazione tra due porzioni di software avviene mediante dispositivi hardware e (forse) altri strati intermedi di software.
- La struttura “logica”, che è un'astrazione della precedente ed è utilizzata per effettuare misure funzionali. Questa struttura mostra che l'utente funzionale (“mittente e/o destinatario designato dei dati”) di una porzione di software sottoposto a misura interagisce con il software attraverso un confine e che il software legge e scrive dati da/a una memoria permanente. In questa astrazione sia l'hardware che il software che permettono queste interazioni non vengono presi in considerazione.

Mentre solo il secondo tipo di struttura è necessario per effettuare una misura funzionale, è utile considerare entrambe, poiché i misuratori devono essere in grado di distinguere tra le due. Inoltre il metodo COSMIC utilizza termini come “strato” e “componente alla pari” con un significato particolare, che deve essere compreso a fondo (questi termini sono utilizzati nell'industria del software con molti significati diversi).

La Figura 2.2.2.1 presenta la struttura fisica di una tipica applicazione aziendale nel contesto di un'architettura a più strati che comprende anche il sistema operativo, il software dei dispositivi di I/O e l'hardware. La Figura 2.2.2.2 presenta la stessa struttura fisica per un semplice esempio di software real-time integrato (“embedded”).

⁷ La conversione di scala tra diversi livelli di granularità viene trattata nel documento “Advanced and Related Topics”.

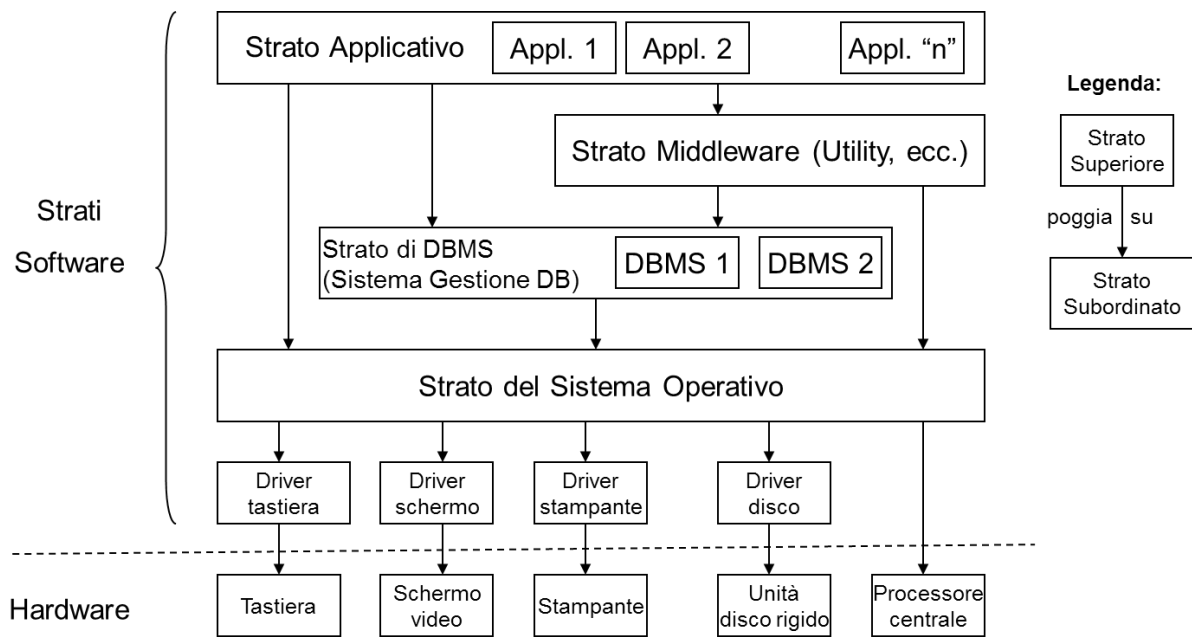


Figura 2.2.2.1 – Tipica architettura software a strati per un sistema informatico aziendale/MIS.

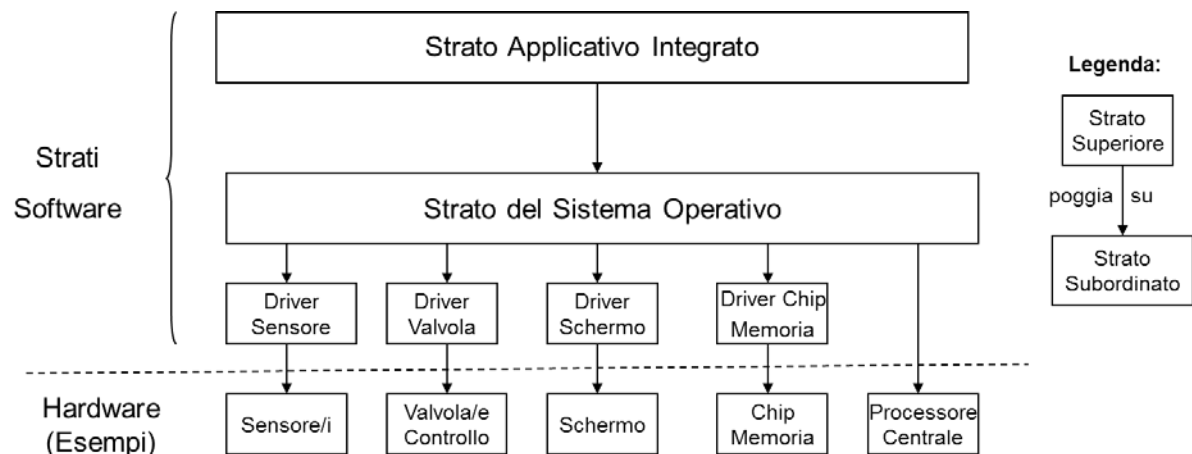


Figura 2.2.2.2 – Tipica architettura a strati per un sistema informatico real-time integrato.

Principio a)

Il software utilizzato da un utente umano è delimitato dalle periferiche di I/O come il mouse, la tastiera, la stampante o il monitor; il software “real time” è delimitato tipicamente da dispositivi come sensori e attuatori. Il software è anche delimitato dall’hardware della “memoria permanente”, come ad esempio il disco fisso o altri tipi di dispositivi che possono essere utilizzati per immagazzinare i dati.

Principio b)

Se la porzione di software che deve essere misurata fa parte di un’architettura a più strati, dovrebbe essere semplice decidere a quale strato appartiene questa porzione. Se il software ha subito un’evoluzione nel tempo, è possibile che gli strati, se esistono, non siano più chiaramente distinguibili. In questo caso il metodo COSMIC definisce delle regole per poter distinguere gli strati

Principio c)

Se i componenti principali di una applicazione aziendale sono componenti alla pari, essi dovrebbero essere misurati separatamente e il metodo COSMIC fornisce le regole per effettuare questo tipo di misurazione (si pensi ad esempio ad una architettura a tre componenti, tutti alla pari, di questo tipo: un componente "front end / interfaccia utente", un componente "regole di business" e infine un componente di servizi per l'accesso ai dati). La capacità di misurare separatamente la dimensione funzionale di ogni componente principale di un software quando vengono eseguiti su piattaforme differenti è molto importante ai fini sia di una stima che della misura.

Qualunque porzione di software in qualunque strato può ovviamente essere scomposta nei suoi componenti a vari livelli (ad esempio a livello di singolo modulo o di classi e oggetti) e il metodo COSMIC può essere utilizzato per misurare la dimensione funzionale del software a ciascuno di questi livelli. I misuratori, però, dovranno definire il locale livello standard di scomposizione, in accordo con l'architetto software o di sistema, al di sotto del livello di scomposizione dei componenti principale, se vogliono permettere il confronto di misurazioni provenienti da fonti differenti

Principio d)

Questo principio richiede che l'ambito di una porzione di software che debba essere misurata sia contenuta interamente all'interno di un singolo strato. Il motivo è che ogni strato assolve ad un specifica funzione e può essere sviluppato utilizzando tecnologie differenti da quelle degli altri strati. Potrebbe essere poco significativo misurare la dimensione funzionale di un software suddiviso in due o più strati, sommando i risultati delle misurazioni di ogni strato e considerando il risultato come la misurazione di una singola entità. Tale risultato potrebbe essere molto difficile da interpretare e/o confrontare con altre misurazioni. Le regole di aggregazione delle misure di porzioni di software appartenenti a strati differenti sono descritte nella sezione del Manuale di Misurazione relativa alle aggregazioni delle misure.

Principio e)

Si assuma che i componenti dell'applicazioni software illustrate nelle Fig. 2.2.2.1 e 2.2.2.2 appartengano a progetti distinti, ognuno con differenti gruppi di sviluppo. In questo caso, per effettuare una misurazione, sarebbe sufficiente definire come ambito l'"intera applicazione". Però, se un'applicazione è costituita da tre componenti alla pari (vedi principio c)), ognuno dei quali utilizza differenti tecnologie e/o viene sviluppato da differenti gruppi di progetto, e se lo scopo della misurazione è la stima dell'impegno necessario, avrebbe senso definire tre ambiti di misurazione separati, uno per ciascun componente. Il risultato della misurazione di ognuno dei tre componenti può essere poi utilizzato come dato di input per un calcolo della stima che tenga conto delle diverse tecnologie e/o delle caratteristiche dei gruppi di lavoro utilizzati.

Principio f)

Allo scopo di chiarire sia questo principio che il successivo principio g), dobbiamo prendere in esame la struttura logica del software che deve essere misurato. Le Figure 2.2.2.3 e 2.2.2.4 mostrano la struttura logica evidenziando l'interazione dell'"utente funzionale" rispettivamente con un applicativo aziendale e con un software real-time integrato.

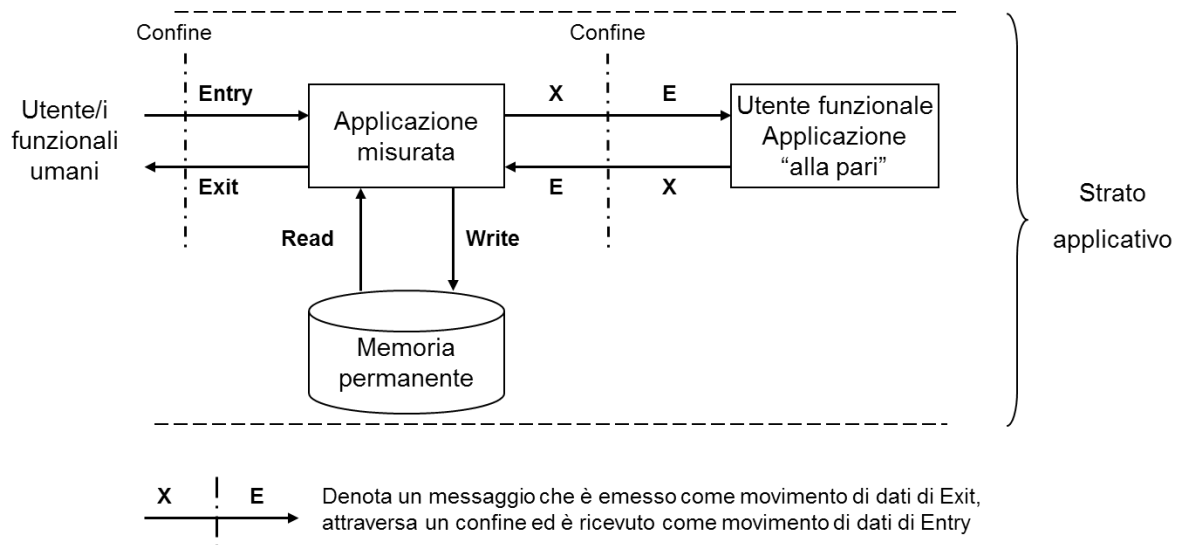


Figura 2.2.2.3 – Applicazione aziendale con sia utenti umani sia un'altra applicazione "alla pari" come propri utenti funzionali (vista logica).

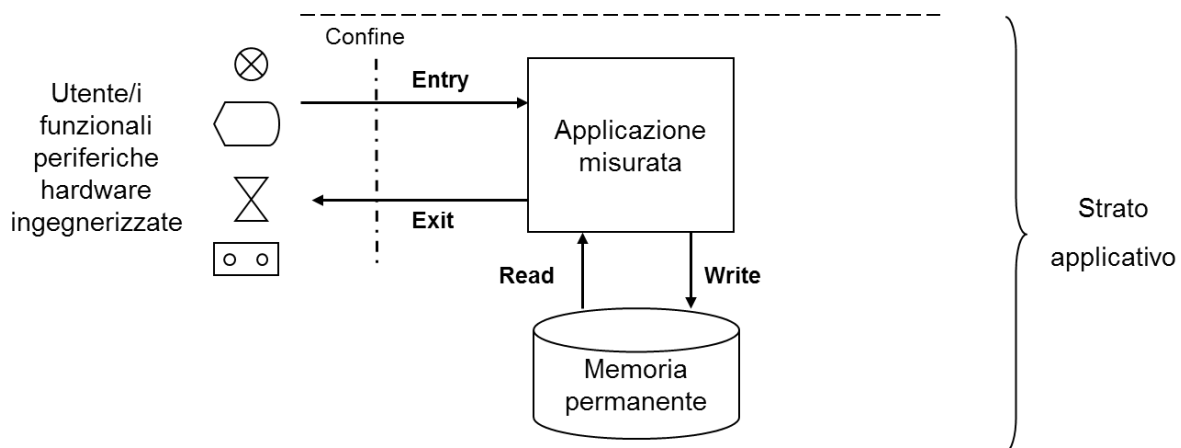


Figura 2.2.2.4 – Applicazione software real-time integrata con varie periferiche hardware ingegnerizzate come suoi utenti funzionali (vista logica).

Si consideri una porzione di software applicativo aziendale da misurare. La Fig. 2.2.2.1 mostra che tra gli "utenti" dell'applicazione si potrebbero considerare anche il sistema operativo, qualunque dispositivo hardware (ad esempio la tastiera o la stampante, ecc..) e gli utenti umani perché tutti loro "interagiscono" con l'applicazione, in maniera diretta o indiretta. Ma non tutti questi (tipi di) utenti vengono specificati nei FUR come mittenti e/o destinatari designati dei dati. Il sistema operativo e i dispositivi hardware permettono questo scambio di dati, piuttosto che esserne i destinatari o i mittenti.

Nel caso di applicazioni aziendali, i FUR descriveranno le funzionalità dell'applicazione solo dal punto di vista degli utenti umani e forse da quello di applicazioni dello stesso livello che inviano o ricevono dati dalla prima. Saranno perciò questo tipo di applicazioni e gli utenti umani ad essere gli unici "utenti funzionali" dell'applicazione, come mostrato nella struttura logica di Fig. 2.2.2.3.

A causa della rigida separazione delle funzionalità in strati mostrata in Fig. 2.2.2.1, i FUR di un'applicazione aziendale possono di solito ignorare ogni strato software o dispositivo hardware come

il sistema operativo o i monitor che permettono l'interazione con i propri utenti funzionali. Di solito non c'è alcun dubbio sull'identità di tali utenti.

Nel caso di software real-time, i suoi FUR descrivono le funzionalità richieste dal punto di vista dei dispositivi hardware che il software deve gestire (sensori, valvole, ecc..). Questi dispositivi saranno quindi gli utenti "funzionali" di questa applicazione, come mostrato in Fig. 2.2.2.4. Come indicato nel Manuale di Misurazione, talvolta i FUR di questo tipo di software vengono scritti riferendosi a più di un tipo di utente funzionale, dando così luogo a differenti funzionalità e, di conseguenza, ad un aumento della dimensione funzionale.⁸

Principio g)

Le Figure 2.2.2.3 e 2.2.2.4 mostrano che gli utenti funzionali interagiscono con il software attraverso un confine per mezzo di due tipi di movimenti di dati (**Entry** e **Exit**). Il software a sua volta scambia dati con l'hardware della memoria permanente per mezzo di altri due movimenti di dati (**Read** e **Write**). Questi movimenti di dati sono dettagliati ulteriormente nel paragrafo 2.2.3

Il "confine" è definito come "un'interfaccia concettuale tra il software sotto esame e i suoi utenti funzionali". Questo confine non deve essere confuso con qualunque confine tracciato per definire o l'ambito della porzione di software che deve essere misurato o uno strato di software.

Il confine permette di fare una chiara distinzione tra ciò che fa parte della porzione di software da misurare (cioè che è dal "lato del software" del confine) e ciò che fa parte dell'ambiente degli utenti funzionali (cioè che è dal "lato dell'utente funzionale" del confine). La memoria permanente non è considerata un utente del software e quindi viene considerata dal "lato del software" del confine.

In Fig. 2.2.2.5 viene mostrata la struttura logica di un'applicazione aziendale che è stata sviluppata come insieme di tre componenti alla pari, come descritto nei principi c) ed e) di cui sopra. Supponendo che i componenti alla pari siano stati sviluppati usando differenti tecnologie, è probabile che venga definito un ambito di misurazione differente per ciascuno dei componenti.

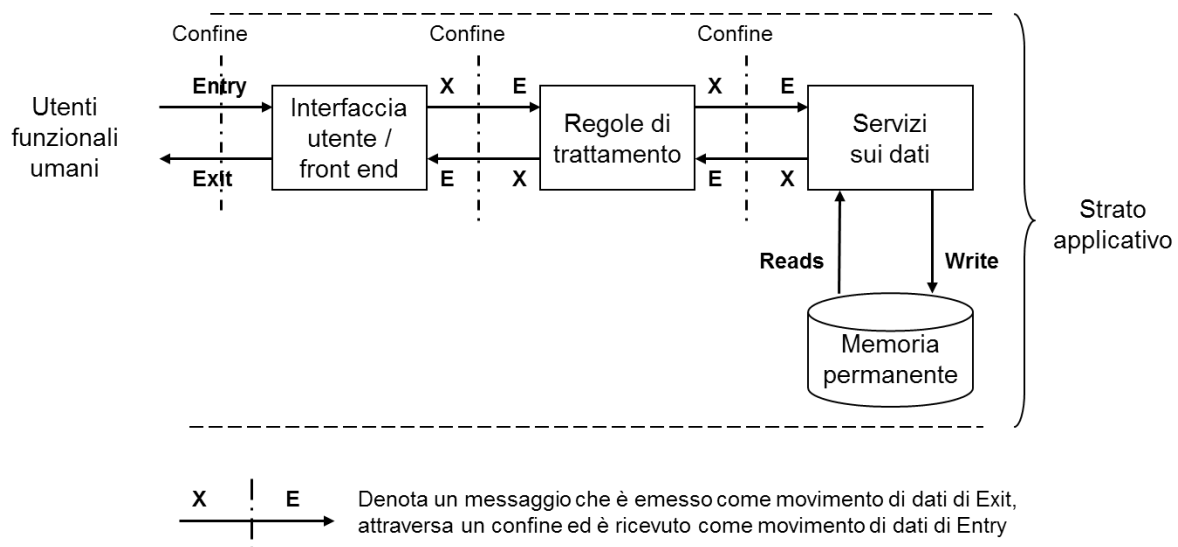


Figura 2.2.2.5 – Applicazione aziendale in cui le componenti "alla pari" devono essere misurate separatamente (vista logica).

⁸ Il sistema operativo può essere considerato un utente funzionale per un'applicazione quando esso è necessario ai FUR dell'applicazione. Un esempio tipico è quello del "clock tick", necessario per innescare l'esecuzione di un processo funzionale.

Dalla Fig. 2.2.2.5 possiamo desumere che il componente “front end / interfaccia utente” ha come propri utenti funzionali sia gli utenti umani che il componente “regole di trattamento”). Ognuno di questi interagisce con il componente per mezzo di Entry ed Exit. La Figura mostra anche che soltanto il componente “servizi sui dati” interagisce con la memoria permanente e che il suo utente funzionale è il componente “regole di trattamento”. Date queste strutture logiche, i FUR di ogni componente possono essere misurati separatamente.

Principi h) e i)

I FUR di un software possono essere espressi a differenti “livelli di granularità”. (Si noti che il concetto di “livello di granularità” riguarda il livello di dettaglio della descrizione di una porzione di software. Questo deve essere distinto dal “livello di decomposizione”, che riguarda la scomposizione del software nei suoi componenti.) Il livello di granularità al quale la misurazione dovrebbe essere effettuata è quello dei processi funzionali (vedi il paragrafo 2.2.3). Nelle fasi iniziali dello sviluppo di un nuovo software, il processo per definire i Functional User Requirement (FUR) inizia tipicamente con una definizione dei requisiti ad alto livello, che viene successivamente perfezionata. I FUR di una porzione di software entro un certo ambito possono quindi esistere a diversi livelli di granularità. Un esempio tipico dell'uso della tecnica dell'“analisi funzionale” per determinare i FUR di una porzione di software può avere come risultato la seguente gerarchia di livelli di FUR.

Una “funzionalità di livello 1”, quando viene analizzata in dettaglio, risulta composta da un certo numero di “funzionalità di livello 2”. Ognuna di esse consiste di “sotto-funzioni di livello 3”, a loro volta costituite da “sotto-sotto-funzionalità di livello 4”, ecc.. Ad un certo livello di questa gerarchia, l'analisi individuerà i singoli processi funzionali (si veda il paragrafo 2.2.3 per ulteriori informazioni sui processi funzionali, per il momento si consideri il processo funzionale solo come l'unità standard della funzionalità che possiamo misurare).

Man mano che l'analisi entra sempre più nel dettaglio, la dimensione funzionale aumenta proporzionalmente perché vengono presi in considerazione sempre più dettagli. (Si noti che questo fenomeno è differente dalla “variazione dell'ambito”, nel quale l'aumento della dimensione funzionale è dovuto all'aumento dell'ambito del software.)

Di conseguenza, per poter confrontare misurazioni provenienti da fonti differenti o per usare le misurazioni in altri tipi di processi, tutte le misurazioni devono essere fatte (o convertite) ad un livello standard di granularità, che potremmo chiamare il “livello di granularità del processo funzionale”. In molti casi, quando lo scopo è misurare la dimensione funzionale di una porzione di software completamente definita, il livello di funzionalità dei processi funzionali è subito evidente.

Chiunque ha familiarità con la misura delle distanze su una mappa, usando scale differenti, dove 1 km è rappresentato sulla mappa da un 1 cm o 1 mm e si possono convertire le distanze da una scala all'altra. Misurando la dimensione funzionale di una porzione di software usando il metodo COSMIC significa, però, avere un unico “livello di granularità del processo funzionale” e un'unica unità di misura. Così, se abbiamo la necessità di comparare misure effettuate a diversi livelli di granularità, dobbiamo introdurre un fattore di conversione per poter ricondurre tutte le misure al livello di granularità standard.

Questi concetti vengono approfonditi ulteriormente nella sezione del Manuale di Misurazione relativa al livello di granularità.

Principio j)

Il problema di misurare una porzione di software ad un livello di granularità maggiore di quello dei suoi processi funzionali di solito nasce solo nelle prime fasi del ciclo di vita di un progetto, mentre i requisiti stanno ancora evolvendosi. In queste circostanze, quando non è possibile misurare al livello di granularità dei processi funzionali, i FUR del software saranno misurati per approssimazione e in seguito convertiti al livello di granularità dei processi funzionali (si veda il Capitolo sulla misurazione nelle prime fasi del progetto del documento “Advanced and Related Topics”).

In definitiva, il Modello del Contesto del Software COSMIC fornisce una serie di concetti e principi, come ad esempio gli strati e i componenti alla pari, l'ambito di misurazione di un software, i suoi utenti

funzionali, i suoi movimenti di dati e il confine, per aiutare a misurare i Requisiti Funzionali Utente, che possono essere definiti a differenti livelli di granularità. Nel corso della fase di Strategia della Misurazione (come descritta nel paragrafo 2.3.1, si applicano i concetti e i principi qui esposti ai FUR del software che deve essere misurato per rispondere a domande come “Che tipo di misurazione si richiede?” o “Come devo interpretare la misurazione?”

2.2.3 Il Modello Generale del Software COSMIC

Avendo interpretato i FUR del software oggetto di misurazione in termini del Modello del Contesto del Software, dobbiamo ora applicare il Modello Generale del Software ai FUR per identificare i componenti delle funzionalità che dovranno essere misurate. Il Modello Generale del Software assume come veri per ogni software che può essere misurato con il metodo COSMIC i seguenti principi generali. Si veda il Glossario per la definizione di tutti i termini⁹.

PRINCIPI – Il Modello Generale del Software COSMIC
a) Il software riceve dati di input dai propri utenti funzionali e produce un output e/o un altro risultato per gli utenti funzionali
b) I requisiti utente funzionali di una porzione di software oggetto di misurazione possono essere mappati su processi funzionali unici
c) Ogni processo funzionale consiste di sotto-processi
d) Un sotto-processo può essere o un movimento di dati o una manipolazione di dati
e) Ogni processo funzionale è innescato da un movimento di dati di Entry da parte di un utente funzionale che informa il processo funzionale che l’utente funzionale ha identificato un evento
f) Un movimento di dati muove un singolo gruppo di dati
g) Un gruppo di dati consiste di un insieme unico di attributi dei dati che descrivono un singolo oggetto d’interesse
h) Esistono quattro tipi di movimenti di dati. Un Entry muove un gruppo di dati all’interno del software da parte di un utente funzionale. Un Exit muove un gruppo di dati fuori del software verso un utente funzionale. Un Write muove un gruppo di dati dal software verso la memoria permanente. Un Read muove un gruppo di dati dalla memoria permanente verso il software
i) Un processo funzionale include almeno un movimento di dati di Entry e un movimento di dati o di Write o di Exit, ovvero include un minimo di due movimenti di dati
j) Come approssimazione per scopi di misurazione, i sotto-processi di manipolazione dei dati non sono misurati separatamente; la/e funzionalità di una qualsiasi manipolazione di dati si assumono come incluse nel movimento di dati a cui essa è associata

Questi principi derivano dalle seguenti considerazioni sul software:

Principi da a) a e)

Lo scopo del software è rispondere ad eventi che avvengono dal “lato dell’utente funzionale” del suo confine, ovvero nel mondo reale dei suoi utenti funzionali. Un’utente funzionale notifica al software un evento e può inviare dei dati relativi all’evento stesso. Il software deve quindi eseguire un’azione utile all’utente funzionale in risposta all’evento. Questa “azione utile” viene definita come “processo funzionale”. Tutti i FUR di un software possono essere espressi come una lista di tipi di eventi e dei corrispondenti processi funzionali che rappresentano la risposta del software a ciascun evento.

⁹ Come evidenziato nel Glossario, ogni metodo di misurazione funzionale identifica “tipi” e non “occorrenze” di dati e funzionalità. Nel testo che segue il suffisso “tipo (di)” viene ommesso quando si citano i concetti base del metodo COSMIC a meno che non sia essenziale distinguere i “tipi” dalle “occorrenze”.

I principi c) e d) affermano che i processi funzionali possono essere considerati come composti da due tipi di sotto-processi, cioè i movimenti di dati e la manipolazione dei dati. Il software può solo muovere e/o manipolare dati.

La Fig. 2.2.3.1 fornisce una rappresentazione grafica dei principi da b) a d) del Modello Generale del Software.

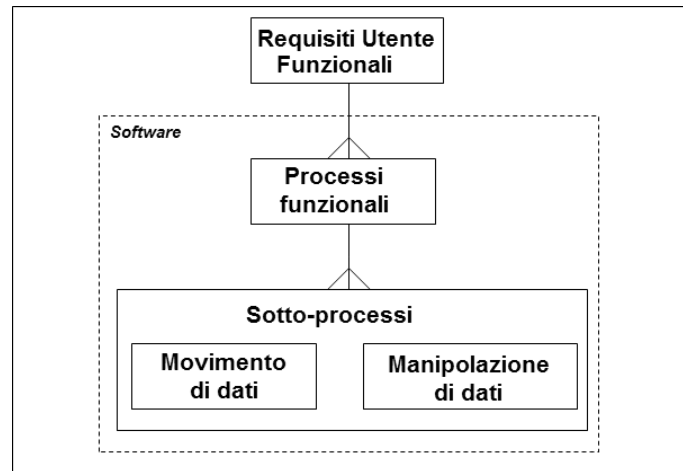


Figura 2.2.3.1 – Struttura dei Requisiti Utenti Funzionali.

Principi f) e g)

Ogni movimento di dati muove uno e un solo gruppo di dati, cioè i dati relativi ad un singolo oggetto di interesse. Un “oggetto di interesse” è qualsiasi “cosa” sia significativa per l’utente funzionale. Un esempio nel dominio delle applicazioni aziendali è il processo funzionale, relativamente semplice, dell’inserimento di un ordine, che tipicamente comprende i seguenti movimenti di dati (gli oggetti di interesse sono scritti tra doppi apici nel seguito):

- Due Entry di gruppi di dati relativi rispettivamente all’“ordine” e alla “riga ordine” (assumendo che si tratti di un ordine con più righe). Il primo degli Entry descrive l’oggetto di interesse “ordine” ed è quello che innesca (cioè fa partire) il processo funzionale.
- Due Read di gruppi di dati relativi rispettivamente al “cliente” e al “prodotto” per verificare che il cliente possa effettuare l’ordine e che i prodotti ordinati siano validi e disponibili.
- Due Write di gruppi di dati relativi rispettivamente all’“ordine” e alla “riga d’ordine” per caricare i dati inseriti nella memoria permanente.
- Una o più Exit di gruppi di dati che contengono per esempio un messaggio di “conferma d’ordine” che comprende il valore totale dell’ordine, istruzioni al magazzino per accantonare ciascun elemento dell’ordine, ecc.

Tutti gli oggetti di interesse di cui sopra sono elementi, reali o concettuali, del mondo reale degli utenti funzionali (umani, in questo caso), di cui il software deve processare i dati. Essi devono essere identificati e distinti in modo da poter identificare i movimenti di dati.

Lo stesso principio si applica anche quando viene misurato il software real-time o embedded, anche se in questi casi molto spesso l’“utente funzionale” e l’“oggetto di interesse” sono virtualmente indistinguibili. Ad esempio, si consideri il caso di un processo funzionale che deve ottenere la temperatura da un sensore e si assuma che il sensore possa comunicare direttamente con il software, così da poter assimilare il sensore ad un utente funzionale del software. In questo caso, il sensore invia un movimento di dati di Entry che probabilmente contiene solo due attributi di dati (ID del sensore e valore della temperatura). Questi due attributi forniscono informazioni sul sensore (come oggetto d’interesse) – anche se in teoria il vero oggetto di interesse è il materiale o l’oggetto di cui il sensore misura la temperatura.

Si consideri ora un processo funzionale molto semplice che misuri la temperatura per mezzo di un sensore e la verifichi rispetto a un valore di riferimento. Il processo funzionale è innescato ad intervalli regolari da un segnale di clock e consiste nei seguenti movimenti di dati:

- Un Entry dal clock che innesca (cioè fa partire) il processo funzionale
- Un Entry dal sensore della temperatura contenente l'ID del sensore e la temperatura rilevata
- Un Read della temperatura di riferimento dalla memoria permanente (assumendo che tale temperatura di riferimento possa essere variata e venga impostata da un altro processo funzionale)
- Un Exit ad un apparato di riscaldamento contenente il segnale di accensione o spegnimento, se lo stato dell'apparato ha bisogno di essere modificato

Si noti che in questo semplice esempio tutti i gruppi di dati, tranne uno, consistono di un solo attributo di dati.

Principio h)

Questo principio afferma che i quattro tipi di movimenti di dati vengono distinti sia in base alla loro sorgente che alla loro destinazione. I movimenti di dati o attraversano il confine tra il software da misurare e i suoi utenti funzionali (Entry e Exit) o muovono dati tra il software e la memoria permanente (Read e Write). Queste relazioni sono presentate in Fig. 2.2.3.2.

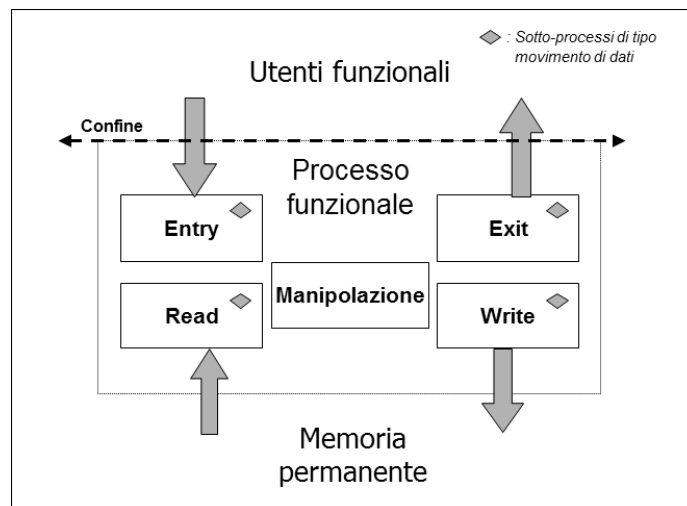


Figura 2.2.3.2 – Componenti di un processo funzionale e alcune loro relazioni.

Principio i)

Questo principio, conseguenza dei principi precedenti, stabilisce che un processo funzionale deve avere almeno due movimenti di dati. Un processo funzionale costituito da un solo movimento di dati e che non esegue nessuna altra azione è inutile. Quindi tutti i processi funzionali devono avere almeno un movimento di dati che li informa dell'occorrenza di un evento (un Entry) e almeno un altro movimento di dati in risposta all'evento accaduto; tale movimento di dati può essere diretto verso l'utente funzionale (un Exit) o verso la memoria permanente (un Write).

Principio j)

Il metodo COSMIC adotta, per poter effettuare le misurazioni, una semplificazione del Modello Generale del Software nei domini software per cui è stato ideato.

In prima approssimazione, in questa versione del metodo di misurazione, i sotto-processi di manipolazione dei dati, illustrati in Figura 2.2.3.2 non vengono considerati separatamente ma sono associati a o sono parte del corrispondente sotto-processo di movimento dei dati (da qui in avanti, per

brevità, verrà usata l'espressione "movimento di dati" piuttosto che "sotto-processo di movimento di dati"). La ragione di questa approssimazione è che le definizioni e i concetti necessari per permettere la misurazione delle manipolazioni di dati non sono ancora stati completamente definiti. Il tipo specifico di manipolazione dei dati che si deve includere in ogni movimento di dati è descritto nella sottosezione del Manuale di Misurazione dedicata alle associazioni tra manipolazioni di dati e movimenti di dati.

Data questa approssimazione, è chiaro che il metodo COSMIC standard è adatto a misurare tipi di software in cui sono presenti molti movimenti di dati (software "movement-rich"), come la maggior parte delle applicazioni aziendali e molto del software real-time, ma non è adatto per applicazioni in cui sono presenti algoritmi complessi o molte manipolazioni di dati (software "manipulation-rich" o "algorithm-rich"). Il Manuale di Misurazione suggerisce di approcciare con cautela la misurazione di porzioni molto piccole di software, o di piccole modifiche al software, dove le assunzioni del principio j) potrebbero non essere più valide.

Il Manuale di Misurazione fornisce anche una procedura per definire estensioni locali al metodo COSMIC che permettono di contare esplicitamente le funzionalità di manipolazione dei dati, se si vuole. In questi casi sono anche previste delle convenzioni particolari per la reportistica della misurazione.

Utilizzando i concetti di cui sopra, le loro definizioni, i principi e le regole del metodo COSMIC, i Requisiti Utente Funzionali estratti dai manufatti di una porzione di software possono essere mappati sul Modello Generale del Software, creandone un'istanza specifica per quel software. Questa istanza conterrà solo gli elementi necessari alla misura della dimensione funzionale, nascondendo tutte le informazioni non rilevanti alla misurazione.

Le regole e i processi di misurazione vengono quindi applicati a questa istanza del Modello Generale del Software per produrre un valore che rappresenta la dimensione funzionale di una porzione di software – si veda il paragrafo 2.3.3 più sotto per le regole di misurazione.

2.3 Panoramica del processo di misurazione COSMIC

Il processo di misurazione del metodo COSMIC consiste di tre fasi distinte ma collegate:

1. la Strategia di Misurazione, in cui si applica il Modello del Contesto del Software
2. la Fase di Mappatura, in cui si applica il Modello Generale del Software
3. la Fase di Misurazione, in cui si ottiene la misura effettiva

2.3.1 Fase di Strategia della Misurazione

Prima di iniziare una misurazione, il misuratore deve accordarsi con il committente della misurazione e deve documentare: a) lo scopo della misurazione, b) lo scopo di ogni porzione di software sottoposto a misurazione, c) gli utenti funzionali di ogni porzione di software e di conseguenza il relativo confine, d) il livello di granularità richiesto per la misurazione. Stabilire chiaramente lo scopo a) della misurazione è di primaria importanza perché determina gli altri parametri b), c) e d). Determinare questi tre parametri è molto spesso un processo iterativo.

(a) Scopo della misurazione

Lo scopo di una misurazione stabilisce perché essa viene effettuata e per cosa verranno usati i risultati. Questo a sua volta aiuta a determinare non solo gli altri tre parametri della Strategia di Misurazione, ma anche, ad esempio, l'accuratezza richiesta per la misurazione (come si può vedere nel Capitolo sulle stime del documento "Advanced and Related Topics", è possibile stimare una dimensione funzionale con una variante di approssimazione del metodo COSMIC standard. Questa variante può essere utilizzata nelle prime fasi del ciclo di vita di un progetto, prima che tutti i requisiti siano completamente definiti e si possano applicare le regole esatte del metodo).

(b) Ambito del software da misurare

L'ambito del software da sottoporre a misura deriva dallo scopo. L'ambito stabilisce quali funzionalità del software saranno comprese nella misurazione (e quali saranno escluse). È dipendente dallo scopo

e può essere suddiviso tra un numero di porzioni di software, ognuno dei quali viene misurato separatamente entro il proprio ambito.

Questa suddivisione dell'ambito è necessaria se il software da misurare è suddiviso in strati, in quanto ogni porzione di software da misurare deve risiedere in un unico strato. Una suddivisione potrebbe essere necessaria se lo scopo della misurazione è quello di fornire una stima e il software da misurare è composto da componenti separati che saranno sviluppati con tecnologie differenti e/o verranno eseguiti su piattaforme differenti e/o saranno sviluppati da differenti gruppi di lavoro. Ogni componente avrà allora il proprio ambito di misurazione.

(c) Utenti funzionali e confine di ogni porzione di software da misurare

Gli utenti funzionali di ogni porzione di software possono essere identificati esaminando i flussi di dati che entrano ed escono dal software come stabilito, in maniera implicita o esplicita, dai suoi requisiti funzionali utente e tenendo conto dello scopo della misurazione. Gli utenti funzionali saranno i mittenti e/o i destinatari designati dei dati.

Nella maggior parte dei casi, l'identificazione degli utenti funzionali risulta ovvia dallo scopo della misurazione e dai FUR. In alcuni casi gli utenti funzionali possono variare al variare dello scopo della misurazione. Nel Manuale di Misurazione è presentato un caso di questo tipo nella sottosezione relativa agli utenti funzionali.

Quando gli utenti funzionali sono noti il confine – l'interfaccia concettuale tra gli utenti funzionali e la porzione di software da misurare – può essere stabilita facilmente.

(d) Livello di granularità della misurazione

Il livello di granularità dei FUR di una porzione di software a cui dovrebbe essere effettuata la misurazione è quello in cui i processi funzionali sono identificati e scomposti nei loro movimenti di dati.

Se lo scopo è misurare i FUR di un software completamente definito, il livello di granularità è di solito evidente, una volta che i processi funzionali siano stati identificati.

Al contrario, nelle prime fasi dello sviluppo di un software, dove lo scopo della misurazione è quello di misurare dei FUR che sono in continua evoluzione, i FUR sono misurati prima che i loro processi funzionali siano tutti identificati e i relativi movimenti di dati definiti. Inoltre, FUR di differenti porzioni di software possono essere definiti a diversi livelli di granularità. In questi casi alcune "unità funzionali" verranno definite localmente, per permettere di effettuare la misurazione al livello di granularità al momento disponibile. Sarà necessario definire anche un metodo per convertire il risultato dal livello di granularità delle "unità funzionali", usate nella misurazione, al livello di granularità in cui i processi funzionali sono identificati e misurati (cioè quello dei Function Point COSMIC). I metodi di conversione sono trattati nel Manuale di Misurazione nella sezione sull'identificazione di un livello standard di granularità e nel documento "Advanced and Related Topics".

Può essere necessario effettuare una o più ripetizioni dopo aver completato almeno una volta tutti i punti precedenti. Ad esempio, se qualche requisito viene dettagliato maggiormente, si può avere la necessità di perfezionare l'ambito del software da misurare.

Una trattazione approfondita e ulteriori esempi su scopo, ambito, utenti funzionali e livello di granularità viene fornita nel Capitolo relativo alla Strategia della Misurazione del Manuale di Misurazione.

2.3.2 Fase di Mappatura

La Fase di Mappatura ha come dati di input i Requisiti Utente Funzionali di ogni porzione di software da misurare, estratti dai relativi manufatti (sia ricavati dalla documentazione prodotta che desunti dal software installato), applicando il Modello del Contesto del Software. Il risultato della Fase di Mappatura è un'istanza specifica del Modello Generale del Software.

I passi per definire un'istanza del Modello Generale del Software sono i seguenti:

- Identificare gli eventi nel mondo degli utenti funzionali a cui il software deve rispondere e di conseguenza identificare i processi funzionali
- Identificare i movimenti di dati (Entry, Exit, Read e Write) per ogni processo funzionale. L'identificazione dei movimenti di dati a sua volta dipende dall'identificazione dei gruppi di dati che vengono mossi

Una trattazione più approfondita con definizioni ed esempi sui concetti e i passi della Fase di Mappatura è presente nel relativo Capitolo del Manuale di Misurazione.

2.3.3 Fase di Misurazione

La Fase di Misurazione prende come dato di input un'istanza del Modello Generale del Software e, utilizzando un insieme definito di regole, produce un valore numerico, direttamente proporzionale alla dimensione funzionale del Modello, basato sul seguente principio:

PRINCIPIO – Il principio di misurazione COSMIC

La dimensione funzionale di una porzione di software è direttamente proporzionale al numero dei suoi movimenti di dati.

Le caratteristiche dell'insieme di regole e processi che guidano la produzione del valore numerico sono le seguenti:

Caratteristica 1 – Unità di misura

L'unità di misura standard, ovvero 1 CFP (COSMIC Function Point), è definito per convenzione come equivalente ad un singolo movimento di dati.

Caratteristica 2 – Proprietà additiva della dimensione entro un ambito di misurazione definito

La dimensione di un processo funzionale è definita come la somma aritmetica del numero dei movimenti di dati che lo costituiscono. Per estensione, la dimensione funzionale di ogni porzione di software, entro un ambito di misurazione¹⁰ in uno strato qualunque, è la somma aritmetica delle dimensioni dei processi funzionali che lo costituiscono.

Caratteristica 3 – Dimensione delle modifiche ad una porzione di software

La dimensione funzionale di ogni modifica ad una porzione di software è per convenzione la somma aritmetica dei movimenti di dati che vengono aggiunti, modificati o rimossi in conseguenza della modifica.

Caratteristica 4 – La minima e massima dimensione di un processo funzionale

In accordo con le caratteristiche di cui sopra, e come già stabilito nel principio i) del Modello Generale del Software, la dimensione funzionale minima di un singolo processo funzionale è di 2 CFP, perché il

¹⁰ Per le regole di aggregazione delle dimensioni funzionali di porzioni di software con differenti ambiti di misurazione, si veda il Capitolo sulla Fase di Misurazione del Manuale di Misurazione.

più piccolo processo funzionale deve avere almeno un Entry (come input) e o un Exit (come output) o un Write (in alternativa, come risultato del processo funzionale).

Siccome una modifica può riguardare un solo movimento di dati, ne segue che la dimensione minima di un cambiamento ad un processo funzionale è di 1 CFP.

Inoltre, in accordo con le caratteristiche di cui sopra, non esiste un limite superiore alla dimensione di un processo funzionale e quindi non esiste nemmeno un limite superiore alla dimensione funzionale di una qualunque porzione di software.

Ulteriori principi e regole e processi dettagliati della Fase di Misurazione, per determinare la dimensione funzionale a partire dai FUR di una porzione di software espressi secondo il Modello Generale del Software, sono presentati nel Capitolo relativo del Manuale di Misurazione e sono sintetizzati nelle sue Appendici B e C.

APPENDICE A – PROCEDURA PER RICHIESTE DI MODIFICA E COMMENTI

Il COSMIC Measurement Practices Committee (MPC) è lieto di ricevere feedback, commenti e, se necessario, richieste di modifica (Change Request) riguardanti la documentazione COSMIC. Questa appendice illustra le modalità per comunicare con il COSMIC MPC.

Tutte le comunicazioni dirette al COSMIC MPC vanno inviate per e-mail al seguente indirizzo:

mpc-chair@cosmicon.com

Feedback e commenti generici informali

Commenti informali e feedback sulla documentazione COSMIC, come per es. difficoltà nella comprensione o nell'applicazione del metodo COSMIC, suggerimenti di miglioramento generale, ecc. dovrebbero essere inviati per e-mail all'indirizzo sopra riportato.

Si prenderà atto dei messaggi e sarà data di norma notifica entro due settimane dalla ricezione. L'MPC non può garantire di mettere necessariamente in pratica tali commenti generici.

Richieste di Modifica formali

Qualora il lettore della documentazione COSMIC ritenga che vi sia un errore nel testo o la necessità di un chiarimento, o che una porzione del testo richieda di essere migliorata, è possibile sottoporre una Richiesta di Modifica (Change Request, CR) formale.

Si prenderà atto delle CR formali e sarà data notifica entro due settimane dalla ricezione. A ogni CR sarà quindi assegnato un numero progressivo e essa sarà fatta circolare tra i membri del COSMIC MPC, un gruppo di esperti a livello mondiale del metodo COSMIC. Il normale ciclo di revisione richiede un minimo di un mese e può richiedere più tempo se la CR si rivela di difficile risoluzione.

L'esito dell'esame della CR può essere "accettata", "rifiutata" o "in attesa di ulteriore discussione" (nell'ultimo caso per es. se esiste una qualche relazione con un'altra CR) e sarà comunicato al mittente non appena possibile.

Una CR formale sarà accettata solo se corredata di tutte le seguenti informazioni.

- Nome, qualifica e organizzazione di appartenenza di chi sottopone la CR.
- Riferimenti di contatto di chi sottopone la CR.
- Data di invio.
- Scopo generale della CR (per es. "occorre migliorare il testo ...").
- Testo esistente che richiede modifica, sostituzione o cancellazione (o chiaro riferimento ad esso).
- Testo aggiuntivo o sostitutivo proposto.
- Spiegazione esauriente del motivo per cui è necessaria la modifica.

Un modulo per sottoporre una CR è disponibile presso il sito web www.cosmicon.com.

La decisione del COSMIC MPC sull'esito dell'esame di una CR e, se accettata, sulla versione del Manuale di Misurazione dalla quale si applicherà la CR è definitiva.

Quesiti sull'applicazione del metodo COSMIC

Il COSMIC MPC si rammarica di non poter rispondere a quesiti relativi all'utilizzo o all'applicazione del metodo COSMIC. Esistono organizzazioni commerciali che possono fornire formazione e consulenza o strumenti di supporto sul metodo. Si prega di consultare il sito web www.cosmicon.com per ulteriori dettagli.