

i Quaderni



1

anno I - marzo 2004

i Quaderni

Pubblicazione a cura del Centro Nazionale per l'Informatica nella Pubblica Amministrazione

Metriche
per lo sviluppo software:
stato dell'arte

Metriche per lo sviluppo software: stato dell'arte

numero 1 - marzo 2004



via Isonzo, 21/b - 00198 Roma
tel. 06 852641
www.cnipa.gov.it

1

anno I
marzo 2004



i Quaderni n. 1 marzo 2004
Supplemento al n. 2/2004
di Innovazione
Periodico
del Centro Nazionale
per l'informatica nella
Pubblica Amministrazione

Anno I
Registrato al Tribunale di Roma
n.523/2003
del 15 dicembre 2003

Direttore responsabile
Franco Tallarita
(tallarita@cnipa.it)

Coordinamento redazionale
Eleonora Piccioni
(eleonora.piccioni@cnipa.it)

Quaderno a cura
di Alessandro Alessandroni

Redazione
Ufficio stampa
Centro Nazionale
per l'Informatica nella
Pubblica Amministrazione
Via Isonzo, 21b
00198 Roma
Tel. (39) 06 852641
Fax (39) 06 85264247
e-mail: ufficio.stampa@cnipa.it

Tutti i numeri dei Quaderni
sono pubblicati nel sito
internet del Cnipa
all'indirizzo:
<http://www.cnipa.gov.it>

Stampa: Stilgrafica srl, Roma

progetto grafico:
Segni di Segni, Roma

i Quaderni

sommario

3

PRESENTAZIONE DELLA COLLANA I QUADERNI

Livio Zoffoli

5

METRICHE PER LO SVILUPPO SOFTWARE: STATO DELL'ARTE

7

DIFFONDERE LA CULTURA SULLA MISURAZIONE
PER DISPORRE DI PRODOTTI MIGLIORI E AUMENTARE LA TRASPARENZA

Livio Zoffoli

9

LA METRICA DEI FUNCTION POINT NELLA PUBBLICA AMMINISTRAZIONE CENTRALE

Alessandro Alessandroni

24

LA METRICA DEI FUNCTION POINT E SUA APPLICABILITÀ
AGLI SVILUPPI SOFTWARE TRADIZIONALI E INNOVATIVI

Domenico Natale

43

L'ASSOCIAZIONE GUFPI-ISMA, I FUNCTION POINT E LE METRICHE FUNZIONALI

Loredana Mancini

54

ANALISI DI TECNICHE DI STIMA DEI COSTI DI SVILUPPO DEL SOFTWARE

Sandro Morasca

74

TAVOLA ROTONDA: IL PUNTO DI VISTA DELLE IMPRESE ICT E DEGLI UTENTI

Presentazione della collana “i Quaderni”

di Livio Zoffoli – Presidente Cnipa

Con questo numero dedicato al convegno “Metriche per lo sviluppo del software: stato dell’arte” il Cnipa dà il via alla pubblicazione della nuova collana “i Quaderni”, che sono particolarmente lieto di poter presentare ai lettori.

Con questa raccolta si apre un ciclo la cui finalità è di promuovere la divulgazione di studi, rapporti e altri documenti tecnici redatti dal Centro nazionale per l’informatica nella pubblica amministrazione.

Attraverso questo strumento di informazione che aggiorna periodicamente sullo stato di attuazione dei principali progetti, il Centro nazionale per l’informatica si propone di veicolare la diffusione di conoscenze ad alto contenuto tecnico e di proporre documenti di riferimento, di ricerca e di aggiornamento su tematiche che il Cnipa approfondisce nel corso delle attività svolte, inerenti il processo di innovazione della pubblica amministrazione.

È a questo titolo che il Cnipa, fin dallo scorso novembre 2003, ha attivato un ciclo di convegni di studio denominato “Dalla informatizzazione alla innovazione” che proseguiranno per tutto il 2004, il cui scopo principale è quello di affrontare una serie di temi specifici di riconosciuta rilevanza e di carattere innovativo al fine di creare uno spazio di confronto, di riflessione e di approfondimento sulle problematiche che scaturiscono dall’utilizzo delle tecnologie della comunicazione e dell’informazione nell’ammodernamento della pubblica amministrazione

Il primo volume de “i Quaderni” contiene gli atti del convegno “Metriche per lo sviluppo del software: stato dell’arte” e raccoglie i contributi dei relatori del Cnipa, della Banca d’Italia e di Sogei intervenuti ai lavori che si sono svolti il 21 novembre 2003 a Frascati presso il Centro Donato Menichella della Banca d’Italia. Tale incontro ha permesso di approfondire il tema delle metriche di sviluppo del software con particolare riferimento alle metriche quantitative, discutendo e verificando l’applicabilità delle metriche tradizionali rispetto alle modalità di sviluppo software innovative oltre che promuovendo un momento di confronto delle esperienze della pubblica amministrazione, del mondo bancario, industriale e accademico in questo campo.

Altri Quaderni sono già in corso di pubblicazione e riguardano il “Vademecum per la realizzazione di progetti formativi in modalità e-learning nelle pubbliche amministrazioni” e il

“Servizio sul protocollo elettronico e documentale in modalità ASP”, allo stesso tempo è in fase di composizione la “Raccolta normativa”, una sintetica selezione di disposizioni che richiama l'attenzione sulle significative e più recenti modifiche normative in materia di informatica nella pubblica amministrazione.

La quantità e, soprattutto, la qualità delle pubblicazioni programmate evidenzia il contributo che la collana de “i Quaderni” potrà conferire allo sviluppo del dibattito sull'informatizzazione pubblica e, più in generale, alla diffusione della cultura informatica nel settore pubblico. Un contributo importante anche per il Cnipa e per la realizzazione della sua missione: contribuire allo sviluppo della pubblica amministrazione, fornendo conoscenze e strumenti che permettano di governare il sistema durante il processo di innovazione tecnologica.

Metriche per lo sviluppo software: stato dell'arte

Nel corso del 2003 il CNIPA ha deciso di avviare il ciclo di convegni di studio “Dalla informatizzazione alla innovazione”, incentrato sull'utilizzo delle tecnologie della comunicazione e dell'informazione finalizzato all'ammodernamento della Pubblica Amministrazione.

I convegni di questo ciclo rappresentano momenti d'incontro e di confronto tra pubbliche amministrazioni, fornitori di soluzioni, mondo accademico, cittadini e imprese su specifici temi del processo di innovazione e digitalizzazione della Pubblica Amministrazione. Ciascun tema viene illustrato da esperti di settore, in grado di presentare una informativa critica degli aspetti tecnici, normativi, sociali ed economici, ed è poi oggetto di una tavola rotonda di discussione e confronto tra le parti.

Il primo convegno del ciclo si è tenuto il giorno 21 novembre 2003 presso il centro Donato Menichella in Frascati, sede della Banca d'Italia, e ha avuto come tema le metriche per lo sviluppo del software. I lavori sono stati introdotti dal presidente del CIPA Cesare Giussani e dal presidente del CNIPA Livio Zoffoli. Gli interventi tecnici, moderati dal professor Aurelio Boari dell'Università di Bologna, sono stati curati da Cnipa, Sogei, Banca d'Italia, Gufpi-Isma, Università dell'Insubria. La tavola rotonda, moderata da Franco Patini, Presidente Anasin e Vicepresidente Federcomin, ha visto la partecipazione, tra i fornitori, di rappresentanti di Engineering, Getronics ed Etnoteam.

L'intervento del Cnipa ha posto l'attenzione sul come e quando la metrica dei punti funzione è stata impiegata nei progetti della pubblica amministrazione centrale, quali risultati ha portato, quali problemi ha consentito di superare, ma anche quali difficoltà si sono riscontrate nel suo utilizzo, e in che modo il CNIPA è intervenuto, nel suo ruolo istituzionale, per indirizzare e supportare le amministrazioni. L'intervento ha poi inquadrato la metrica dei punti funzione nel più vasto panorama delle metriche per la qualità del prodotto e del processo applicate allo sviluppo e alla manutenzione del software nella pubblica amministrazione, illustrando quanto in passato è stato fatto e quali sono le prospettive future, con riferimento anche al monitoraggio dei progetti, allo studio della produttività, alla realizzazione di benchmark.

L'intervento della Sogei ha affrontato, nella veste di fornitore/sviluppatore di applicazioni informatiche, il tema dell'applicabilità della metrica dei punti funzione agli sviluppi software in ambienti innovativi quali i sistemi ERP, i CRM, i packages personalizzabili, i sistemi autore generativi, ecc. E' stato detto che, per tali ambienti innovativi, diverse ricerche sono in corso per determinare misure alternative al punto funzione per la determinazione dei costi.

L'intervento ha illustrato l'uso di misure funzionali anche per analisi qualitative standardizzate, ad esempio per correlare la difettosità riscontrata nell'ambiente reale ai punti funzione in esercizio. L'intervento ha approfondito tali aspetti e delineato le condizioni di successo di applicazione dei punti funzione, prendendo in considerazione gli aspetti organizzativi e la competenza delle persone che effettuano i conteggi.

L'intervento della Banca d'Italia si è focalizzato sull'esperienza dell'Istituto sia in progetti di sviluppo di applicazioni informatiche condotti internamente sia come cliente di fornitori esterni. L'intervento ha evidenziato alcune problematiche connesse all'uso della metrica dei punti funzione nella determinazione dei costi di progetto, quali ad esempio il fenomeno del "backfiring dai costi", e ha posto l'attenzione sulle caratteristiche del progetto che, a parità di dimensione dell'applicazione, influenzano i costi di sviluppo.

L'intervento del Gufpi-Isma è consistito essenzialmente nella presentazione degli scopi, della struttura e delle attività dell'associazione stessa. Sono stati descritte le finalità costitutive dei vari comitati tecnici, e in particolare del gruppo di lavoro CPC (Counting Practices Committee) che ha come scopo il miglioramento della formulazione delle regole di conteggio dei punti funzione, del gruppo di lavoro SBC (Software Benchmarking Committee) interessato all'approfondimento delle tecniche di standardizzazione usate per confrontare diverse performance nello sviluppo software con particolare riferimento alle produttività e costi unitari, e del gruppo SMC (Software Measurement Committee) che si occupa degli aspetti di misurazione da un punto di vista non funzionale, trattando aspetti organizzativi quali piani di misurazione, iniziative di miglioramento del processo di sviluppo del software, tool metrici e misurazione delle performance.

L'intervento dell'Università dell'Insubria si è focalizzato a livello più teorico sulle diverse tecniche di stima dei costi di sviluppo del software. Dopo una loro classificazione, si è analizzato criticamente tali tecniche e sono state evidenziati i loro punti di forza e debolezza, in modo tale da fornire elementi per prendere decisioni su quali tecniche sono più facilmente e utilmente applicabili in specifiche applicazioni.

La tavola rotonda ha messo a confronto il punto di vista di vari fornitori sull'utilità e sull'applicabilità delle metriche nell'ambito delle commesse di sviluppo e/o manutenzione del software. Sulla base delle rispettive esperienze, e con particolare riferimento a quanto realizzato con la pubblica amministrazione in qualità di cliente, i fornitori hanno discusso le problematiche connesse alle misure di quantità e di qualità del software, alla produttività nei progetti di sviluppo, all'utilizzo di strumenti di rilevazione, storicizzazione, consolidamento e benchmarking, all'applicazione delle misure negli aspetti contrattuali, ad esempio nella definizione dei costi unitari e delle penali.

1. Diffondere la cultura sulla misurazione per disporre di prodotti migliori e aumentare la trasparenza

Dall'intervento del Presidente del Cnipa Livio Zoffoli ad apertura dei lavori del convegno

Con il primo convegno "Metriche sullo sviluppo software: stato dell'arte", tenutosi nella sede di Vermicino della Banca d'Italia, si è inaugurato il ciclo di convegni di studio del Cnipa.

Esiste un forte interesse comune sul tema della misurazione del software e della sua qualità, il convegno risponde appunto all'esigenza di approfondire una tematica molto sentita dagli operatori e dalle imprese: è una questione che riguarda la conoscenza, ma che per altro verso comporta anche un volume di spesa molto rilevante.

Il Centro nazionale con questo ciclo di convegni si propone di trattare numerosi temi, che riteniamo vitali per lo sviluppo del Paese e in particolare per la digitalizzazione della pubblica amministrazione ed il miglioramento dei servizi offerti a cittadini e imprese.

Il 10 dicembre è già stato fissato un nuovo appuntamento per un evento dedicato alla qualità della postazione di lavoro informatizzata dei dipendenti pubblici; si terrà presso la sede dell'Inail, che ringrazio per la collaborazione offerta per la sua realizzazione.

Altri convegni sono in via di definizione per l'inizio del prossimo anno e riguarderanno: archiviazione ottica e gestione dei flussi documentali, e-learning, open source e riuso, documentazione giuridica on line, brevettabilità del software, accessibilità all'informazione e ai servizi.

Il Cnipa è coinvolto in qualità di promotore tecnico di questi temi di straordinaria rilevanza e si impegna ad incentivare l'uso di queste tecnologie nell'interesse del Paese, per la sua crescita nei diversi settori bancario, imprenditoriale, privato, oltre che fra i cittadini in generale.

I cittadini e le imprese devono accorgersi che lo Stato sta cambiando e in quest'azione di trasparenza occorre far conoscere quali sono gli strumenti che vengono utilizzati per realizzare il cambiamento; come devono essere messi in grado di valutare la nostra azione.

È dunque fondamentale essere in condizione di misurare e valutare la qualità. L'incontro di oggi ci permette di approfondire le tecniche della misurazione, in altra occasione vedremo di affrontare anche i criteri per definire la qualità.

Venendo più in particolare al tema esaminato nel primo convegno, le amministrazioni pubbliche fanno un rilevante ricorso all'acquisto di prodotti software sul mercato, a volte con procedure concorsuali frequentemente valendosi di trattative private, ma nel complesso si tratta di prodotti che vengono acquisiti con i criteri più disparati; nessuna amministrazione usa gli stessi criteri di un'altra nel fare le acquisizioni.

Nello svolgimento della sua attività istituzionale, che è quella di esprimere pareri sui progetti delle pubbliche amministrazioni, il Cnipa si trova quotidianamente di fronte a questa

disomogeneità e incontra significative difficoltà nel far capire alle amministrazioni a quali pericoli si espongono sottovalutandone la portata; per ovviare alla mancanza di conoscenza e per superare le difformità nei criteri adottati, il nostro compito deve essere di dare alle amministrazioni, al mercato, delle linee guida.

È stato istituito un tavolo di lavoro, coordinato dal Cnipa, cui partecipa il mercato oltre al mondo accademico e a tre importanti amministrazioni, che si sono mostrate disponibili a dare la propria collaborazione.

L'obiettivo è individuare una metodologia per valutare la qualità. Quando una amministrazione sottoporrà al Cnipa un progetto per ottenere un parere di congruità, non dovrà tenere conto solo del prezzo, ma dovrà essere in grado di misurare i prodotti, considerando gli aspetti della qualità e allo stesso tempo anche il manuale dei livelli di servizio, che già è stato elaborato e pubblicato (vedi sito www.cnipa.it - pubblicazioni - quaderni - anno 2002, n° 7).

Per fare questo non c'è bisogno di formare nell'ambito della pubblica amministrazione altri tecnici informatici, quanto piuttosto di disporre di persone che conoscano gli strumenti a disposizione e che sappiano governare i processi.

Il manager pubblico deve possedere questi requisiti professionali: se riusciremo in tale impegno, avremo conseguito l'obiettivo di ottenere prodotti superiori, che durino nel tempo e che forniscano un servizio migliore.

2. La metrica dei Function Point nella pubblica amministrazione centrale

Intervento di Alessandro Alessandroni - Cnipa (alessandroni@cnipa.it)

(con la collaborazione di Francesco Grasso)

2.1 Abstract

La pubblica amministrazione centrale italiana ha storicamente fatto grande ricorso, per lo sviluppo e la gestione dei propri sistemi informativi, a forniture di servizi di sviluppo e manutenzione di software.

Il CNIPA ha visibilità e un ruolo attivo di indirizzo e di supporto sui progetti della pubblica amministrazione centrale: nel presente articolo si illustra come e quando la metrica dei punti funzione è stata impiegata, quali risultati ha portato, quali problemi ha consentito di superare, ma anche quali difficoltà si sono riscontrate nel suo utilizzo, e in che modo il CNIPA è intervenuto a indirizzare e supportare le amministrazioni.

L'articolo inquadra poi la metrica dei punti funzione nel più vasto panorama delle metriche per la qualità del prodotto e del processo applicate allo sviluppo e alla manutenzione del software nella pubblica amministrazione, illustrando quanto in passato è stato fatto e quali sono le prospettive future, con riferimento anche al monitoraggio dei progetti, allo studio della produttività, alla realizzazione di benchmark.

Riguardo alle prospettive future, vengono infine illustrate alcune iniziative condotte dal CNIPA, anche sul tema "metriche alternative ai punti funzione per sviluppi software in ambienti innovativi".

2.2 Autori

Alessandro Alessandroni

(alessandroni@cnipa.it)

Responsabile Area Osservatorio del mercato ICT del CNIPA. Ha svolto attività lavorativa, didattica e di ricerca nel campo dei sistemi di supporto alle decisioni e dei sistemi informativi in organizzazioni complesse. Ha pubblicato articoli e testi scientifici su queste tematiche. Dopo 20 anni di esperienze lavorative in imprese industriali e finanziarie, opera da alcuni anni nella Pubblica Amministrazione, prima in AIPA e ora in CNIPA.

Francesco Grasso (grasso@cnipa.it)

Analista dell'Osservatorio del mercato ICT del CNIPA. La sua area di competenza specifica include le problematiche di produzione del software, le metodologie di progetto, le architetture software, le nuove tecnologie e strumenti per lo sviluppo applicativo, i sistemi ERP, il software open source. Ha 10 anni di esperienza professionale, maturata sia in imprese industriali che nella pubblica amministrazione.

2.3 Contesto

Nella realizzazione e nella gestione dei propri sistemi informatici, le pubbliche amministrazioni italiane si trovano frequentemente a intraprendere progetti di sviluppo o di manutenzione di software.

In termini economici, tali attività rappresentano

una percentuale significativa del totale delle spese delle amministrazioni centrali in campo informatico: come riferimento, dai dati del consuntivo 2001, 244 milioni di euro circa (il 13% del totale spese IT) è destinato a sviluppo o manutenzione di software ad hoc.

I progetti di sviluppo e manutenzione di software vengono condotti:

- basandosi su risorse interne all'amministrazione;
- affidando tutte o alcune delle attività previste dal progetto all'esterno.

Nella pubblica amministrazione centrale la prima modalità è minoritaria, e specifica soltanto di alcune situazioni particolari, quali la presenza di gruppi di competenza (tecnologica, progettuale) interni, la necessità di un forte controllo sul processo di produzione, l'esistenza di alti requisiti di sicurezza e confidenzialità nei sistemi da sviluppare o mantenere, o la concentrazione "storica" interna della conoscenza su particolari tipologie di applicazioni informatiche facenti parte del "core business" dell'amministrazione.

Al contrario, l'affidamento a società terze delle attività di sviluppo e manutenzione di software

è la soluzione assolutamente predominante: si stima che nella pubblica amministrazione centrale, nel suo complesso, più dell'80% di tali attività siano condotte affidandole a fornitori di servizi esterni.

Il contratto di fornitura può essere di tipo:

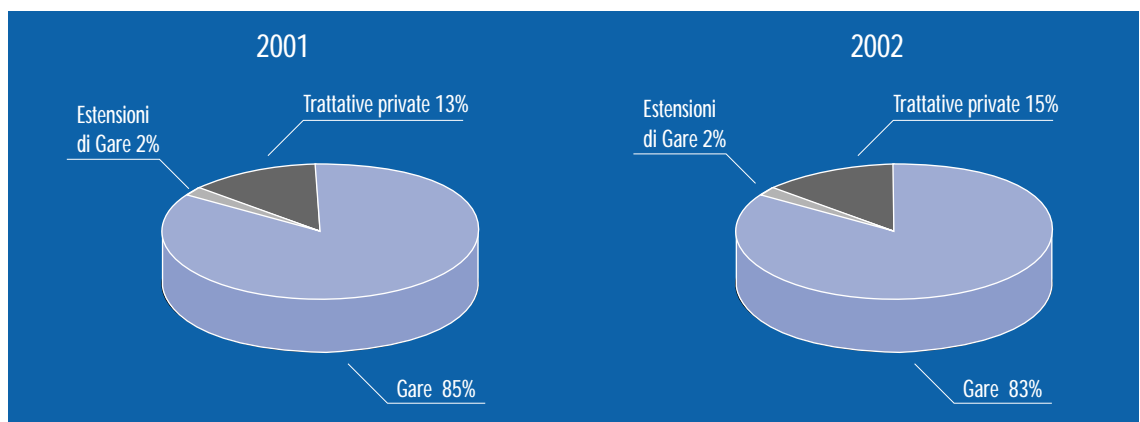
- accordo chiuso (o "a corpo", o "chiavi in mano");
- accordo aperto (o "a tempo e spesa", o "time rental").

Nel primo caso, il prodotto finale richiesto è completamente definito in termini di funzioni, prestazioni e modalità operativa. Possono inoltre essere presenti vincoli sugli ambienti d'esercizio (hardware e/o software di base), sui linguaggi di sviluppo, sulle caratteristiche della documentazione, eccetera.

Nel secondo caso, il contratto riguarda in pratica la fornitura non di un prodotto bensì di prestazioni professionali, delle quali il cliente può richiedere essenzialmente la qualità e la continuità.

Tipicamente, l'affidamento a società esterne avviene tramite gara pubblica, in minor misura a seguito di trattativa privata tra l'amministrazione e uno o più fornitori. Prendendo come

Figura 1: ripartizione percentuale, in termini economici, tra gare e trattative private



base di osservazione i progetti di sviluppo e manutenzione evolutiva nella PAC su cui il CNIPA ha reso un parere nel corso del 2001 e del 2002, si osserva la ripartizione percentuale di figura 1.

Come si nota, l'83% della spesa della PAC in sviluppi e manutenzione evolutiva nel corso del 2002 è relativa a gare pubbliche (85% nel 2001), il 15% è relativa a trattative private (13% nel 2001), mentre il 2% è relativo ad estensioni di gare pubbliche (affidamento di ulteriori richieste di servizi relativi a un progetto di sviluppo all'aggiudicatario della gara bandita per il progetto stesso).

In termini assoluti e non percentuali, la ripartizione relativa all'anno 2002 è la seguente:

- Gare: € 62.007.365
- Estensioni di gare: € 1.382.039
- Trattative private: € 11.502.012
- Totale: € 74.891.415

Mentre quella relativa all'anno 2001 è la seguente:

- Gare: € 103.713.842
- Estensioni di gare: € 1.934.648

- Trattative private: € 16.382.529
- Totale: € 122.031.018

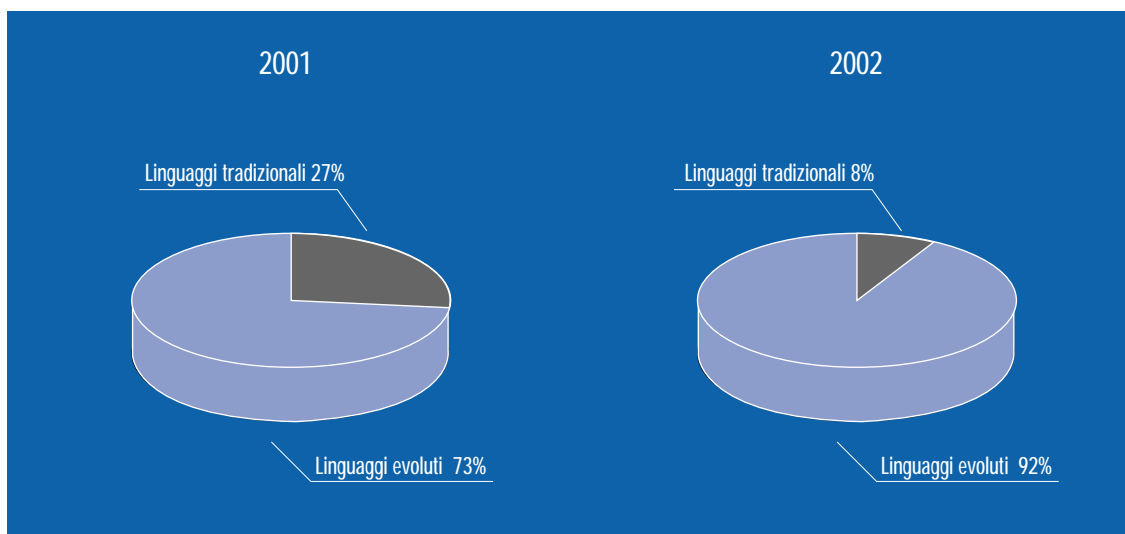
Dal punto di vista tecnologico, in passato era assolutamente preponderante, nei progetti di sviluppo e di manutenzione applicativa, l'uso del linguaggio di programmazione Cobol e delle piattaforme legacy (soprattutto mainframe IBM, ma anche Bull, Siemens, ecc.).

Nel corso degli ultimi anni, viceversa, il peso relativo del Cobol si è drasticamente ridotto nei confronti di linguaggi di programmazione più evoluti (Java, C++, Visual Basic, ecc.).

Allo stesso tempo, il ruolo dei sistemi legacy si è andato limitando: l'uso di tali sistemi è tuttora significativo come piattaforma di produzione e per applicazioni di back office, ma ad essi si sono andati sempre più affiancando piattaforme aperte basate su sistemi operativi Unix, Windows e, in percentuale minore, Linux.

Prendendo ancora come base di osservazione l'insieme di progetti della PAC su cui il CNIPA ha reso un parere negli ultimi anni, la situazione rilevata è quella di figura 2: la percentuale d'utilizzo di linguaggi tradizionali (Cobol) dimi-

Figura 2: ripartizione percentuale tra i linguaggi di programmazione



nuisce dal 27% del 2001 all'8% del 2002, mentre la percentuale d'utilizzo di linguaggi evoluti cresce dal 73% del 2001 al 92% del 2002.

Riguardo al parco applicativo attualmente in esercizio, si prende a riferimento ancora il dato del consuntivo del 2001. In quell'anno la PAC conteggiava un patrimonio applicativo di 493.064 punti funzione, di cui:

- 153.593 punti funzione (31%) in linguaggio Cobol,
- 339.471 punti funzione (69%) in linguaggi evoluti o di alto livello.

A questi numeri va addizionata la porzione di patrimonio applicativo ancora espressa in KLOC, ovvero 366.829 migliaia di linee di codice, di cui:

- 251.758 KLOC (68%) in cobol o altri linguaggi di basso livello,
- 115.071 KLOC (32%) in linguaggi evoluti o di alto livello

Accanto al cambiamento del linguaggio di programmazione prevalente, negli ultimi anni sono avvenute altre rilevanti evoluzioni (tecnologiche, ma anche riguardanti aspetti metodologici) nei progetti di sviluppo applicativo. Tra queste, le più significative sono state le seguenti:

- introduzione dell'analisi e della progettazione object-oriented;
- passaggio dall'architettura client-server all'attuale architettura web-based, col sempre più frequente ricorso a middleware (es. application server), all'impiego del browser quale client dell'applicazione, alla fruizione delle applicazioni attraverso portali;
- utilizzo di sistemi di sviluppo integrati (RAD, CASE), di metodologie di analisi standard e di strumenti automatici o semi-automatici per la generazione del codice;
- passaggio dall'approccio "progettare e sviluppare da zero un software ad hoc" all'approccio

"personalizzare e parametrizzare un pacchetto commerciale", con l'individuazione della migliore soluzione di base "off the shelf" ("dallo scaffale", ovvero disponibile sul mercato), l'analisi dello scostamento dai requisiti, lo sviluppo di estensioni, il riuso eccetera;

- integrazione delle applicazioni e delle basi di dati all'interno di sistemi più complessi quali datawarehouse, sistemi di gestione documentale e workflow, ERP, sistemi di e-learning, eccetera.

2.4 Le metriche del software

La misurazione del software rappresenta un campo nuovo e relativamente giovane dell'ingegneria del software (circa 30 anni) e consiste nella quantificazione degli attributi dimensionali e qualitativi dei prodotti software e delle attività correlate al loro sviluppo (processi).

Le metriche trovano applicazione nelle aziende e nei team di sviluppo per gestire, controllare e migliorare i processi di produzione e sono utilizzate da alcuni acquirenti per governare le forniture di sviluppo e di manutenzione del software, per definire quantità e costo dei prodotti richiesti e definire i relativi profili di qualità.

La dimensione del software

La metrica più diffusa per misurare la dimensione del software prodotto è denominata Function Point Analysis (FPA) proposta nel 1979 da Allan J. Albrecht, tesa a misurare i punti funzione (function points - FP), ossia le funzionalità espresse da un programma e rilevabili dal punto di vista dell'utente finale.

Tale metrica si è diffusa in Italia a partire dai primi anni '90 in sostituzione della metrica delle Linee di Codice (Lines of Code - LOC) basata sul conteggio delle istruzioni eseguibili e di quelle relative alla dichiarazioni dei dati.

I limiti dell'uso delle LOC per analisi di produttività e di stima derivano principalmente dal fatto che le linee di codice misurano in realtà soltanto la "lunghezza" dei programmi software in termini di istruzioni rilasciate.

Le linee di codice dipendono pertanto dal linguaggio di programmazione utilizzato: ogni linguaggio possiede una propria espressività che consente di ottenere un dato risultato con un numero di LOC inversamente proporzionale al livello di maturità del linguaggio stesso. Inoltre le LOC possono essere calcolate solo in fasi molto avanzate del ciclo di vita del software, comportando nelle stime un elevato margine di soggettività e aleatorietà.

La FPA è aggiornata costantemente da un gruppo internazionale di società e organizzazioni che utilizzano questa metrica, denominato IFPUG (International Function Point User Group), rappresentato in Italia dal GUFPI-ISMA (Gruppo Utenti Function Point Italia-Italian Software Metrics Association). L'IFPUG redige periodicamente un manuale delle regole di conteggio, arrivato alla release 4.1.1)

I punti funzione trovano largo impiego nei si-

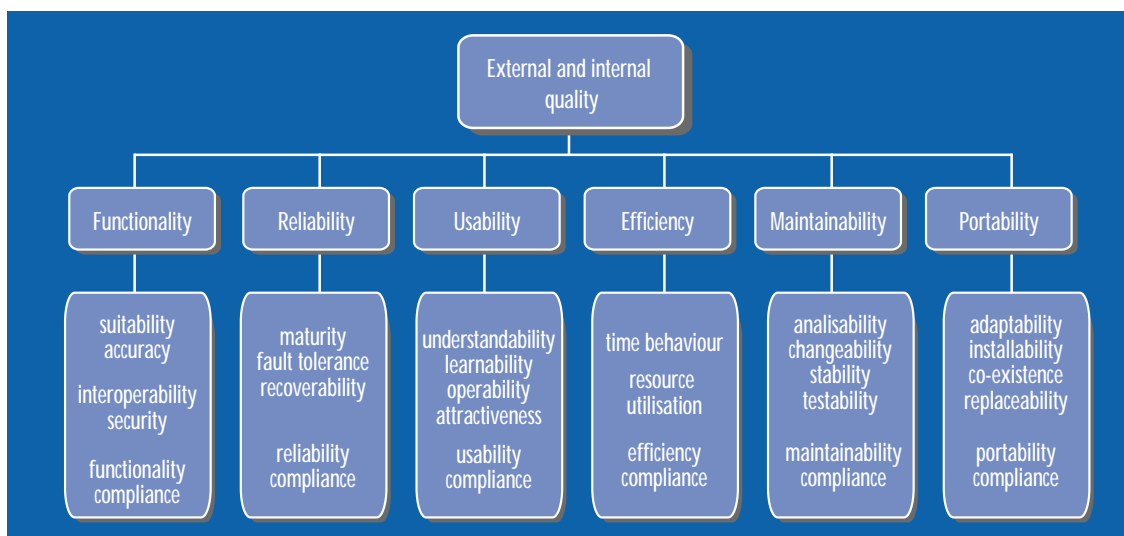
stemi di misurazione, lungo tutto il ciclo di sviluppo del software, in relazione ad aspetti di produttività, qualità ed economici. Di seguito vengono riportati, come esempio, indicatori basati sui punti funzione:

- produttività:
 - in sviluppo: ore di sviluppo/FP sviluppati;
 - in manutenzione: numero di FTE/FP in esercizio
 - velocità di rilascio: FP sviluppati/tempo calendario
- qualità:
 - difettosità: difetti nel primo anno/FP
 - copertura dei test: numero di casi di test/FP
 - documentazione: pagine/FP
- economico:
 - costo per fp sviluppato: costo totale/FP sviluppati
 - costo in manutenzione: costo manutenzione/FP in esercizio
 - valore del portafoglio: FP totali x costo unitario

La qualità dei prodotti

La norma ISO/IEC 9126, pubblicata nella sua prima versione nel 1991, ha definito il modello

Figura 3: Caratteristiche della qualità ISO 9126



dei requisiti qualitativi del prodotto software. La norma distingue tre famiglie di caratteristiche che influenzano tre diversi aspetti della qualità di un software, corrispondenti a diversi punti di vista della qualità:

- qualità interna, misurabile sulle caratteristiche intrinseche come il codice sorgente;
- qualità esterna, misurabile nel comportamento del prodotto a seguito di sollecitazioni dall'esterno (ad es. in un test);
- qualità in uso, misurabile solo durante l'utilizzo effettivo da parte dell'utente, nel reale contesto d'uso.

Secondo tale modello, aggiornato nel 2000, le caratteristiche relative alla qualità interna ed esterna sono raggruppabili in 6 "caratteristiche" e in 27 "sottocaratteristiche" (vedi figura 3). Le caratteristiche della qualità in uso sono invece quattro: efficacia, produttività, sicurezza, soddisfazione.

Le norme ISO 9126 non si limitano a definire un modello di qualità ma forniscono anche indicazioni sulle metriche da utilizzare per misurare le caratteristiche di qualità.

Nelle parti 2,3 e 4 della ISO/IEC sono definite alcune possibili metriche per misurare, rispettivamente, la qualità del prodotto esterna, interna e in uso, con indicazione delle caratteristiche e sottocaratteristiche alle quali fanno riferimento, della formula e tecnica di misura.

La qualità dei processi

La norma ISO 9000-3 del 1994 (e la successiva revisione del 1998) ha adattato al mondo del software il dettato della ISO 9001 per fornire una guida apposita per la realizzazione del sistema di qualità e facilitare la strutturazione e la razionalizzazione dei processi del Ciclo di Vita del software.

La revisione dello standard 9000-3:2000 fa se-

guito alla emissione della serie ISO 9000:2000 ("Vision 2000").

La norma copre le fasi di:

- specifica dei requisiti (aspetti commerciali, tecnici, manageriali e legali),
- progettazione (analisi, architettura, disegno e codifica),
- consegna (archiviazione, di packaging, di duplicazione, di installazione e distribuzione), manutenzione (correttiva, evolutiva ed adattativa).

2.5 Utilizzo delle metriche del software nella PA

Per governare efficacemente le forniture di servizi di sviluppo e manutenzione applicativa affidate a società esterne, le amministrazioni pubbliche da sempre hanno sentito la necessità di metriche riconosciute e condivise con i fornitori, con le quali quantificare i servizi oggetto di forniture e definire correttamente i massimali di gara nel caso di remunerazione a corpo e non a tempo e spesa.

L'uso di opportune metriche permette infatti di oggettivare maggiormente l'oggetto dei contratti, minimizzando le ambiguità, le incomprensioni, e contenendo le differenze tra valori iniziali e finali entro limiti ragionevoli. Consente inoltre di stabilire in modo preciso costi e schedulazioni del progetto, di verificare lo stato di avanzamento del progetto stesso, e di specificare modalità di controllo della qualità del software prodotto.

Le metriche ideali, o meglio le metriche di cui le amministrazioni hanno bisogno, devono rispondere ai seguenti requisiti:

- essere correlate con l'impegno di progetto (effort);
- riflettere la complessità;
- comunicare valore;
- essere semplici da calcolare (non richiedere grande sforzo per l'utilizzo).

L'assenza di metriche adeguate è un problema sentito: acquistare a corpo un prodotto/servizio che non si riesce a quantificare correttamente significa, infatti, acquistare disponibilità/capacità produttiva con un trasferimento di rischio sulla parte più debole tra il cliente e il fornitore. Fin dall'inizio della propria attività l'AIPA ha favorito, con azioni di indirizzo e di supporto, l'adozione da parte delle amministrazioni di metriche del software, quantitative (metodo dei punti funzione secondo IFPUG) e qualitative (principalmente ISO 9126 e ISO 9000-3), durante l'intero ciclo di vita:

- studio di fattibilità,
- richiesta di parere/gara/contratto,
- monitoraggio,
- collaudo,
- manutenzione evolutiva e correttiva.

L'azione di indirizzo si è esplicitata principalmente attraverso le linee guida per gli studi di fattibilità, la normativa sul monitoraggio e le indicazioni contenute nei pareri.

Le azioni di supporto hanno riguardato la consulenza diretta e l'erogazione di corsi di formazione sulla materia.

Fino ai primi anni '90, nella pubblica amministrazione veniva utilizzata, nel campo del software, la metrica delle linee di codice sia per le attività di sviluppo che di manutenzione.

La prima indicazione dell'AIPA di adottare la metrica dei punti funzione è contenuta nel parere del dicembre 1993 relativo al Quarto Atto esecutivo della Convenzione di Concessione con SOGEI per lo sviluppo del Sistema Informativo del Ministero delle Finanze. Per le prestazioni professionali riguardanti la realizzazione dei programmi elaborativi era prevista, dal contratto, la corresponsione da parte dell'Amministrazione di un importo determinato sulla base del numero delle linee di programma realizzate e

del costo unitario di ogni singola linea di programma prodotta. Nel considerato del parere si osservava al riguardo: *“che tale meccanismo, ancorché siano previsti sistemi di controllo per la verifica dei livelli di produttività virtuale media in base a specifiche clausole di salvaguardia per l'Amministrazione, non consente di conoscere, a priori, il costo del servizio richiesto e, pertanto, per il futuro, anche per tale aspetto si rende necessario adottare una metodologia diversa, per la corretta valutazione dell'impegno necessario alla realizzazione dei programmi, quale, ad esempio, il sistema dei "punti funzione".*

Nel corso dei dieci anni intercorsi da quella data, la metrica dei punti funzione è stata adottata progressivamente dalla quasi totalità delle pubbliche amministrazioni centrali. Nel 2002 sono stati emessi 38 pareri relativi a contratti di pubbliche amministrazioni centrali per servizi di sviluppo e manutenzione software basati sulla metrica dei punti funzione, per un totale di 260.435 punti funzione e un valore complessivo di 74.891.415 euro.

Il CNIPA ha partecipato attivamente alle iniziative del GUFPI-ISMA; ha messo a disposizione delle amministrazioni esperti, alcuni certificati IFPUG, e ha promosso numerose iniziative di formazione sul tema tra le quali si segnalano i seminari:

- Introduzione all'uso dei Function Points ([www.cni-pa.gov.it/site/itIT/Le_Activit%
c3%a0/Formazione/Catalogo_formation/Seminari/Seminario/Introduzione_all'uso_dei_Funtion_Points.html](http://www.cni-pa.gov.it/site/itIT/Le_Activit%c3%a0/Formazione/Catalogo_formazione/Seminari/Seminario/Introduzione_all'uso_dei_Funtion_Points.html))
- Metodi e metriche per la stima dell'impegno in ambienti innovativi: aspetti teorici e casi di studio.

Analoga attenzione è stata rivolta alla misura degli aspetti qualitativi. Già a partire dal 1994, anno di pubblicazione della norma ISO 9000-

3:1994, in un parere relativo alla reingegnerizzazione del sistema dell'Area Pensioni dell'INPS, fu posta la condizione di adottare "nell'esecuzione del progetto metodi e parametri di misurazione della qualità per le operazioni di controllo e test ispirati alle norme: ISO 9126 per i prodotti, e 9000-3 per il processo del fornitore". In successivi pareri sono state indicate alle amministrazioni le metriche da utilizzare per misurare le caratteristiche di qualità della ISO 9126 e definire i valori soglia contrattuali. Per ciascun indicatore proposto sono stati indicati la formula, la tecnica di misura e il valore soglia.

Ad esempio, per la misura della caratteristica affidabilità / sottocaratteristica maturità è stato proposto come indicatore la densità di errori sul prodotto (dopo i test finali) ovvero la difettosità residua dopo il rilascio rapportata alla dimensione della applicazione (misurata in punti funzione).

La formula relativa è la seguente:

n° errori applicativi in un anno/volume in FP del prodotto in esercizio.

La tecnica di misura: verifiche su archivi di log e interviste all'utenza. I valori di soglia suggeriti variano di volta in volta, in relazione alla criticità della applicazione. Ad esempio, nei pareri resi nel corso dell'ultimo anno, tale soglia è variata da un minimo di 0,5% (ovvero 5 ogni 1000 PF) nel parere 135/2002 a un massimo del 12% (relativo al primo trimestre dopo la messa in produzione) nel parere 40/2003. La verifica dei profili di qualità definiti contrattualmente è stata oggetto delle attività di monitoraggio.

Recentemente l'ultima caratteristica del modello ISO 9126, la portabilità, ha assunto particolare rilievo nell'ambito della pubblica amministrazione per favorire il riuso dei pro-

grammi informatici di proprietà della amministrazioni. La recente direttiva del Ministro per l'innovazione e le tecnologie in materia di sviluppo e utilizzazione dei programmi informatici da parte delle pubbliche amministrazioni, al punto 7, recita: "Al fine di favorire il riuso dei programmi informatici di proprietà delle amministrazioni, nei capitolati o nelle specifiche di progetto dovrà essere previsto, ove possibile, che i programmi sviluppati ad hoc siano facilmente portabili su altre piattaforme".

2.6 Iniziative Cnipa in corso

Raccolta e valutazione di riferimenti

Sulla tematica della produttività e dei costi dei progetti di sviluppo e/o di manutenzione evolutiva, l'Osservatorio del Mercato del CNIPA impiega come riferimenti le principali fonti internazionali in materia.

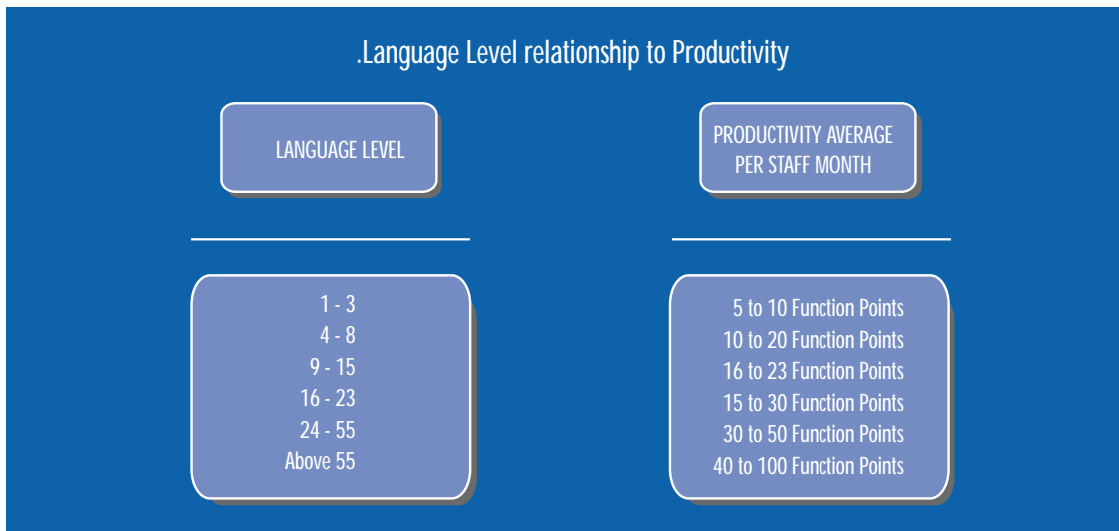
Tra questi, si segnalano:

- gli studi di Capers Jones, derivanti dall'attività di consulenza del noto analista presso grandi società americane, e regolarmente pubblicati sotto forma di volume in edizioni aggiornate. Tra i risultati più noti dei suoi studi, la classificazione dei linguaggi di programmazione in più "livelli", corrispondenti a diversi valori di produttività in sviluppo (vedi tabella a fianco).

Come esempio, Capers Jones inquadra il linguaggio Cobol nel livello 3, il C++ e il Java nel livello 6, il Visual Basic nel livello 11, il Clipper nel livello 17, il Perl nel livello 15 e lo HTML nel livello 22.

- le analisi di Gartner Group, che sul tema della produttività e dei costi dello sviluppo software, nonché della manutenzione, compie attività di benchmarking, con confronti con riferimenti di caratteristiche paragonabili oppure di "best in class". Negli ultimi anni Gartner ha com-

Figura 4: Relazione del livello del linguaggio con la produttività



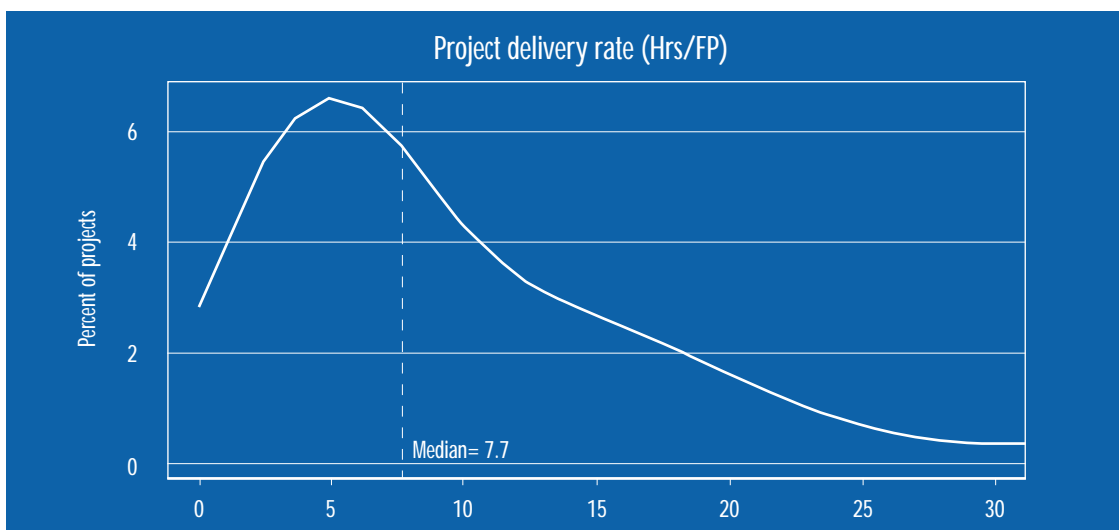
più valutazioni di produttività e di costi di sviluppo per alcune amministrazioni centrali: i risultati di tali valutazioni figurano tuttora tra i riferimenti CNIPA.

- i benchmark di ISBSG (International Software Benchmarking Standard Group). Lo ISBSG è un'organizzazione senza fini di lucro, con sede in Australia, che conduce progetti

di studio e ricerche sui temi dello sviluppo e della manutenzione del software, delle metriche e della produttività nei progetti informatici. Periodicamente, tale organizzazione rilascia, dietro pagamento, i risultati di rilevazioni di produttività in sviluppo software compiute su scala mondiale.

La più recente edizione della rilevazione (ISBSG

Figura 5: Distribuzione della produttività secondo ISBSG

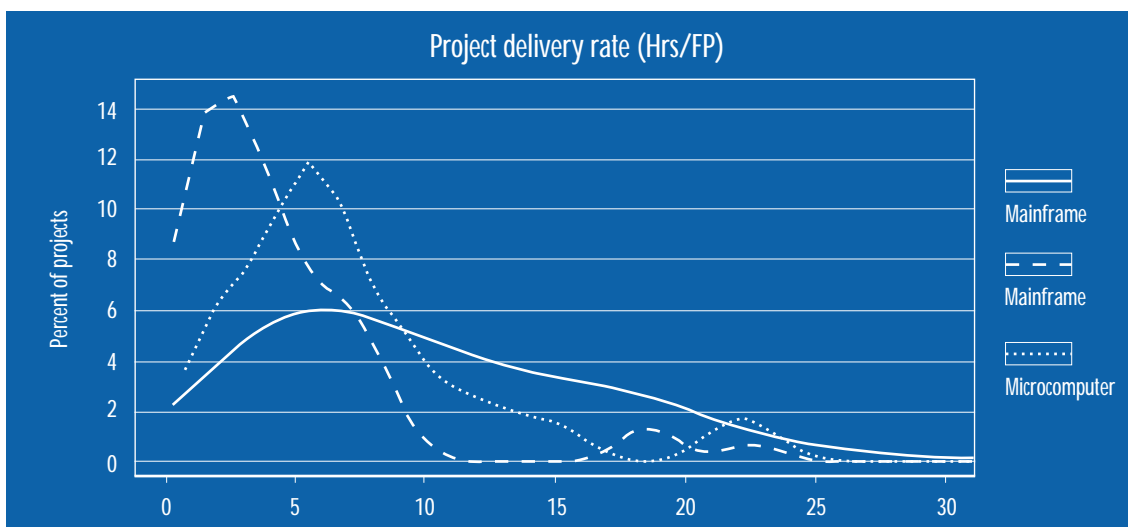


Data Disk release 8) comprende i dati di oltre 2000 progetti, di cui vengono date le caratteristiche e i parametri di produttività quali il Project Delivery Rate (ore di lavoro per punto funzione) e lo Speed of Delivery (punti funzione rilasciati al mese).

La rilevazione ISBSG include un buon numero di analisi statistiche. Si riporta ad esempio la distribuzione del Project Delivery Rate Rate (ore di

lavoro per punto funzione) nei progetti in funzione della piattaforma di sviluppo, classificata in mainframe, midrange, o microcomputer (PC). Pur riconoscendo grande importanza e significatività alle fonti di cui sopra, l'Osservatorio ha sentito l'esigenza anche di un riferimento "interno" maggiormente focalizzato sulle caratteristiche, dimensionali e strutturali, della pubblica amministrazione italiana.

Figura 6: Produttività in sviluppo per piattaforma



Una fonte in grado di rispondere all'esigenza di cui sopra è costituita dai progetti di sviluppo e/o manutenzione della PAC su cui il CNIPA ha reso un parere. Limitatamente agli ultimi tre anni, è stata compiuta una rilevazione sistematica nell'archivio dei pareri, estraendo da questi i dati progettuali significativi, in particolare:

- numero di punti funzione da sviluppare o mantenere;
- linguaggio di programmazione e piattaforma di sviluppo;
- tipologia di fornitura (gara o trattativa privata);
- percentuale del ciclo di vita del software oggetto di fornitura;

- costo del punto funzione;
- eventuali requisiti di qualità del software da sviluppare.

La serie storica dei dati elencati viene attualmente usata dal CNIPA per valutazioni di congruità, analisi di scostamento, studi sul peso dei singoli parametri e sull'andamento temporale di alcuni indicatori.

Tra i vantaggi di questa fonte, l'estrema omogeneità dei campioni, che permette comparazioni e aggregazioni dei dati impossibili ad esempio nell'archivio di ISBSG.

L'uso di tale fonte, ovviamente, non prevarica né esclude il ricorso alle altre. Al contrario, si ritiene necessario comunque una pluralità di

fonti, perché l'uso esclusivo dei riferimenti interni porterebbe inevitabilmente a una "autoreferenzialità" dell'analisi: in altre parole, considerando quale unico riferimento i pareri resi in passato, le analisi di congruità economiche del CNIPA non potrebbero che appiattirsi nel tempo sulla media delle valutazioni precedenti.

Studio di adattamenti/evoluzioni della metrica dei Punti Funzione

Sia in risposta a richieste di singole amministrazioni, sia nel corso dei suoi compiti istituzionali, l'Osservatorio del Mercato CNIPA ha preso in esame la problematica delle metriche per sviluppo applicativo e manutenzione nei nuovi ambienti tecnologici, con particolare riferimento alla realizzazione di siti web, di datawarehouse, di sistemi ERP e di applicazioni di reportistica.

In questo esame, si è scelto di non concentrarsi sulle modalità di conteggio dei punti funzione o su eventuali estensioni di questa metrica ai nuovi ambienti: tali aspetti sono infatti allo studio di organizzazioni maggiormente focalizzate sul tema, quali ad esempio il GUFPI, di cui peraltro il CNIPA è tra i soci onorari. Al contrario, la questione su cui ci si è concentrati è la relazione tra la dimensione in punti funzione dell'applicazione da realizzare nei nuovi ambienti e l'impegno effettivamente necessario. Si ritiene infatti che, fatte salve le eventuali integrazioni o estensioni allo studio del GUFPI e di altri soggetti, la metrica dei punti funzione in teoria sia applicabile anche ai nuovi ambienti di sviluppo, in quanto per definizione essa consente di misurare il contenuto funzionale, dal punto di vista dell'utente, di un'applicazione a prescindere dalla tecnologia con la quale l'applicazione stessa viene realizzata.

Tuttavia, ammesso che il conteggio porti a un risultato corretto dal punto di vista teorico, sembra poi non esserci una diretta relazione tra tale numero e l'impegno di realizzazione in termini di giorni uomo, e dunque il costo finale del progetto. Nel seguito, vengono illustrati alcuni esempi su aree tematiche specifiche.

Uso di componenti di base

Ipotizziamo che il numero di punti funzione di un'applicazione da realizzare, conteggiati con il metodo tradizionale IFPUG a partire dai requisiti utente, sia 100.

L'applicazione, tuttavia, viene realizzata non sviluppando da zero il codice, bensì acquistando un pacchetto di base disponibile sul mercato, e poi adattandolo ai requisiti mediante una parametrizzazione e una personalizzazione.

In tal caso, una percentuale dei 100 punti funzione conteggiati non viene effettivamente sviluppata, bensì è inclusa nelle funzionalità offerte dal pacchetto di base. L'impegno necessario per l'adattamento del pacchetto non è dunque dipendente dal numero di punti funzione dell'applicazione finale, ma dipende da altri fattori quali ad esempio la modularità del pacchetto di base, il suo grado di adattabilità, la disponibilità o meno di ambienti editore integrati eccetera.

Più in generale, poiché la modalità di realizzazione di applicazioni informatiche sta evolvendo, come detto, dallo sviluppo da zero all'integrazione e all'estensione di componenti di base, il calcolo dell'impegno di realizzazione non può prescindere dal riuso di funzionalità già disponibili, il che consente risparmi sia nello sviluppo del codice (si parla in questo caso di riuso "tecnico") che nell'analisi (si parla in questo caso di riuso "funzionale").

Siti Web

La realizzazione di siti web, sia per uso interno all'organizzazione (intranet) che esterno (internet), è un'attività che generalmente viene suddivisa in tre sotto-attività tra loro distinte, anche dal punto di vista concettuale:

- studio e realizzazione della logica applicativa;
- studio e realizzazione dell'interfaccia grafica;
- redazione e composizione dei contenuti informativi.

La prima attività è assimilabile a uno sviluppo applicativo di tipo tradizionale. L'impegno necessario si può perciò valutare a partire dal numero di punti funzione con le formule e i modelli disponibili in letteratura.

La seconda attività è di tipo creativo, non strutturata, e può essere quantificata in termini di giorni di impiego di adeguate figure professionali (grafico, web publisher, esperto multimediale, ecc.) di cui l'Osservatorio del Mercato CNIPA rileva sistematicamente le tariffe di riferimento.

La terza attività prevede l'impiego di professionalità di tipo giornalistico, e comprende compiti redazionali quali la raccolta e la strutturazione di informazioni, la composizione di testi eccetera. Normalmente viene quantificata in base alla dimensione (in cartelle) dei documenti realizzati, oppure in termini di giorni persona impiegati e delle tariffe giornaliere di riferimento.

Datawarehouse

Nel settore della realizzazione di datawarehouse esistono studi e linee guida, ad esempio del GUFPI, che definiscono in dettaglio le opportune modalità di conteggio dei punti funzione. Anche in questo caso, si ritiene che il numero di punti funzione conteggiato con tali modalità, ancorché corretto dal punto di vista teo-

rico, non sia però necessariamente in relazione con l'impegno di progetto in termini di giorni uomo. Ciò perché le sotto-attività più onerose di questa tipologia di progetti, ovvero:

- definizione e gestione dei metadati;
- pulizia, correzione e integrazione dei dati degli archivi di partenza;

non corrispondono a esplicite funzionalità offerte all'utente, e dunque non vengono generalmente incluse nel numero di punti funzione. Inoltre, in questa tipologia di progetti gran parte del lavoro non avviene mediante scrittura di codice, ma con l'ausilio di strumenti grafici o semiautomatici, ad esempio tool di ETL (extraction, transformation, load), strumenti interattivi per l'OLAP, generatori di datamart e di specifiche viste utente.

Di fatto, i maggiori fornitori di sistemi datawarehouse utilizzano, nei loro progetti, stime dell'impegno basate su benchmarking interni e consuntivi di precedenti progetti. In pratica tali fornitori:

1. esaminano alcune caratteristiche strutturali del progetto in esame,
2. sulla base di queste caratteristiche classificano il progetto,
3. ricercano, nell'archivio interno, progetti "simili" (della stessa classe) da cui estrarre dati di riferimento (durata, produttività, eccetera).

Peraltro, i fornitori di cui sopra, in occasione di contratti con la pubblica amministrazione, effettuano anche dimensionamenti del progetto usando i punti funzione. Ciò perché hanno la percezione che per le P.A. tale metrica sia di affidabilità contrattuale maggiore, e in qualche modo sia "rassicurante". Ma il conteggio dei punti funzione viene eseguito in modo che il risultato sia allineato con la stima dei costi effettuata con metodi interni. Si tratta spesso, come si suole dire, di un "backfiring dai costi".

ERP

Nel campo dei sistemi ERP (Enterprise Resource Planning), l'Osservatorio del Mercato CNIPA ha condotto uno studio della durata di alcuni mesi, raccogliendo documentazione tecnica, confrontando le opinioni di analisti e di esperti, e tenendo in considerazione le posizioni dei principali fornitori e integratori di prodotti (tra le società intervistate, SAP, Oracle, JDEdwards-PeopleSoft, KPMG, Bull, Byte, Formula, ESA, Zucchetti, Kyneste).

A seguito di tale studio, il CNIPA ha rilasciato delle linee guida per le amministrazioni per l'acquisizione di servizi di implementazione di sistemi ERP. Tali linee guida illustrano, tra l'altro, tecniche di stima per il dimensionamento di un progetto d'implementazione di questa tipologia di sistemi. Si rimanda al relativo documento (http://www.cnipa.gov.it/site/_contentfiles/00464200/464286_quaderni_9.pdf) per maggiori informazioni.

Sulla base delle tecniche di stima illustrate nel

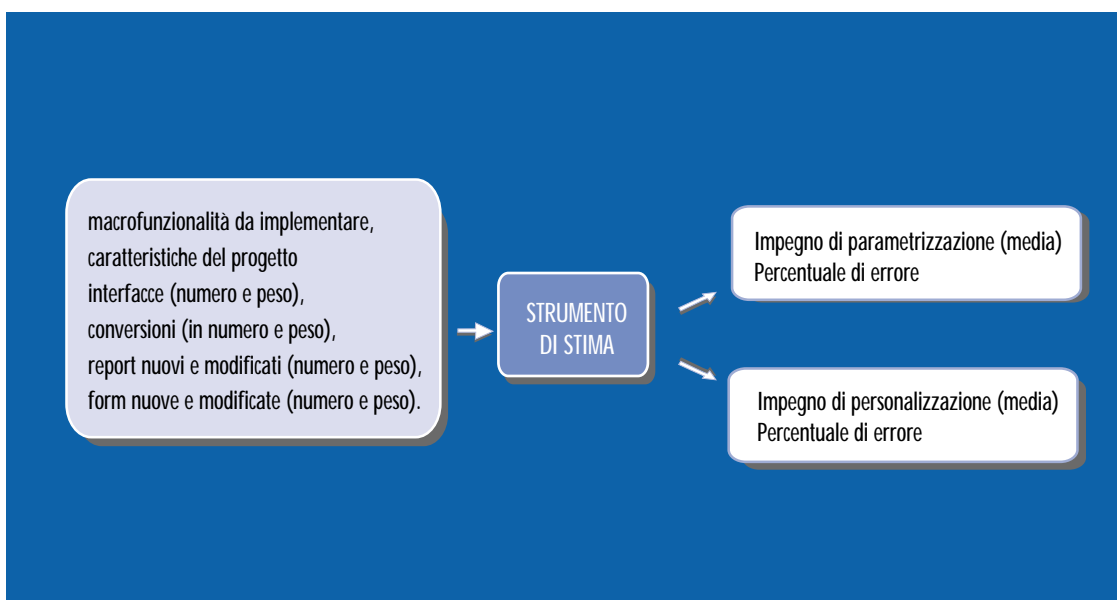
documento citato, è stato realizzato un tool informatico che automatizza il processo di stima, accettando in ingresso:

- le macrofunzionalità da implementare,
 - le caratteristiche del progetto (fattori di rischio e variabili di stima),
 - le interfacce (in numero e peso),
 - le conversioni (in numero e peso),
 - i report nuovi e modificati (numero e peso),
 - le form nuove e modificate (numero e peso),
- e fornendo in uscita:

- una stima media per l'impegno di progetto, in termini di giorni uomo,
- una percentuale di errore su tale stima.

Lo strumento fornisce anche impegni parziali, cioè suddivide la stima tra impegno per la parametrizzazione e impegno per le varie sotto attività di personalizzazione (interfacce, conversioni, report, form). E' inoltre indipendente dalla specifica tecnologia impiegata (non viene definito il particolare ERP da utilizzare nel progetto).

Figura 7: Stima dell'impegno in progetti ERP



Il tool può essere impiegato non solo per stimare i costi di progetto, ma anche per effettuare simulazioni (what if) e valutare quali alternative progettuali fanno aumentare i costi, e quali invece consentono risparmi. Fino a oggi, è stato utilizzato internamente al CNIPA, ma è stato anche sperimentato da alcune amministrazioni interessate alla materia (si cita ad esempio la regione Umbria).

Sistemi di reportistica

Numerose pubbliche amministrazioni centrali impiegano (o prevedono di impiegare in futuro), per la produzione sistematica di reportistica, applicazioni informatiche basate su prodotti commerciali, quali ad esempio Business Object.

Anche in questo caso, per stimare l'impegno di realizzazione, la metrica dei punti funzione non sembra adatta: il conteggio porta a un risultato teoricamente corretto, corrispondente al contenuto funzionale dell'applicazione, ma che non ha in genere relazione diretta col numero di giorni uomo occorrenti per realizzare l'applicazione stessa. Tipicamente, applicazioni di questo tipo vengono infatti realizzate non scrivendone il codice, bensì utilizzando degli appositi "ambienti autore", con interazione normalmente di tipo grafico, del prodotto di reporting.

Al momento questa problematica è oggetto di studio da parte dell'Osservatorio. Si è deciso, quale primo passo d'indagine, di compiere una rilevazione all'interno della PAC sui progetti (in corso o già ultimati) di realizzazione di sistemi di reportistica. Oggetto della rilevazione, le caratteristiche del progetto, la produttività in sviluppo, la durata, il costo giornaliero del mix di risorse messo in campo dal fornitore (oppure dalla stessa amministrazione nel caso di sviluppo interno).

Le amministrazioni centrali coinvolte nella rilevazione sono Inail, Ministero della Salute, Ministero della Giustizia, Ministero dei Trasporti, Ministero Affari Esteri, Istat, Ministero della Difesa, Ministero dell'Interno, Ministero del Tesoro (Consip). Alcune tra queste hanno già fornito i dati richiesti, altre devono ancora restituire il modello di rilevazione.

Allo stesso tempo, ci si è confrontati sull'argomento con i fornitori, in primo luogo Business Object, sia in qualità di vendor della tecnologia sia come implementatori di soluzioni. L'approccio che al momento sembra più ragionevole, per stimare l'impegno di implementazione, prevede i seguenti passi:

- decomporre il sistema di reportistica da realizzare in una serie di oggetti elementari,
- definire una tassonomia (classificare gli oggetti per tipologie),
- estrapolare l'impegno elementare medio per l'implementazione delle varie tipologie di oggetti elementari,
- calcolare l'impegno complessivo come sommatoria degli impegni elementari al netto di una percentuale di riuso.

Ciò significa usare gli oggetti elementari di cui sopra quale metrica dimensionale effettiva per la stima dell'impegno di sviluppo.

Al momento si sta verificando l'efficacia di questa metodologia sulla base dei dati risultanti dalla rilevazione in corso.

Conclusioni e raccomandazioni

L'esperienza compiuta in questi anni dal CNIPA sul tema delle metriche per il software porta a formulare, in sintesi, le seguenti raccomandazioni.

Alle amministrazioni si raccomanda di:

- utilizzare, nei progetti di sviluppo e manutenzione applicativa, metriche sia qualitative che quantitative;
- per conseguire i risultati attesi in termini di efficacia e minore conflittualità, utilizzare le metriche di cui sopra con attenzione e competenza;
- formare personale sul tema delle metriche, mediante percorsi formativi adeguati ed eventualmente esami di certificazione.

Si suggerisce altresì ai fornitori una maggiore attenzione alla problematica delle metriche,

nell'ottica che esse rappresentano uno strumento utile, nella gestione del progetto, anche dalla parte di chi esegue la fornitura.

Infine, si invita il mondo accademico e della ricerca a contribuire a diffondere cultura sull'argomento "metriche", a recepire le difficoltà del mondo degli utenti e a fornire soluzioni, e soprattutto a proporre metodologie e strumenti che rendano semplice l'applicazione delle metriche ai tipici progetti della pubblica amministrazione.

3. La metrica dei Function Point e sua applicabilità agli sviluppi software tradizionali e innovativi

Intervento di Domenico Natale – SOGEI (dnatale@sogei.it)

Abstract

Le diverse possibilità di misurare oggetti e fenomeni sono a tutti evidenti. Nel mondo del software il Function Point (FP) è una misura standard della dimensione funzionale, misura che riduce l'astrattezza del prodotto, fornendo dei riferimenti precisi da valutare.

Negli ambienti tradizionali, ed in alcune condizioni negli ambienti innovativi, la grandezza dell'applicazione risulta altamente correlata con l'impegno profuso per le attività effettuate durante il ciclo di produzione del software ed il FP in questi casi è largamente utilizzato sia per aspetti dimensionali che di costo. In alcuni ambienti innovativi (ERP, CRM, Packages personalizzabili, Sistemi autore generativi, ecc.) i FP relativi all'insieme complessivo delle funzioni fornite dall'applicazione possono essere ottenute sia da automatismi specifici che attraverso apposite realizzazioni o personalizzazioni ex-novo, riducendo, nel primo caso, la correlazione tra i FP finali complessivi e l'impegno impiegato nella realizzazione del prodotto. Per tali ambienti innovativi diversi ambienti di ricerca sono attivi per determinare nuove soluzioni alternative al FP per la determinazione dei costi, mentre parallelamente l'IFPUG sta definendo alcune nuove linee guida di riferimento. In entrambi gli ambienti tradizionali e innovativi, comunque, l'eventuale disponibilità di una misura funzionale è utile per analisi qualitative standardizzate, ad esempio per correlare la difettosità riscontrata nell'ambiente reale ai FP in esercizio. Nell'articolo verranno approfonditi tali aspetti e saranno anche delineate le condizioni di successo di applicazione dei FP, compresi gli aspetti organizzativi, gli skill delle persone che effettuano i conteggi e la loro certificazione CFPS (Certified Function Point Specialist).

1. Introduzione

L'interesse verso i Function Point è iniziato in Italia agli inizi degli anni '90, negli USA negli anni '80, anche se il metodo è stato definito in IBM da Allan Albrecht nella seconda metà degli anni '70. La misura più utilizzata era in quegli anni la Linea di codice, ma il FP, grazie anche ad associazioni come l'IFPUG e l'ISO (International Organization for Standardization), è diventata nel tempo una misura di dimensione standard. In Italia la metrica è utilizzata in varie organizzazioni e l'associazione italiana GUFPI/ISMA è il riferimento degli utenti dei Function Point.

I FP misurano la dimensione delle funzionalità di un'applicazione, indipendentemente dalla tecnologia utilizzata e dall'effort impiegato, secondo le convenzioni standard riconosciute internazionalmente.

In alcuni ambienti italiani già nell'inizio degli anni '90 si sosteneva che bisognava dare una mag-

giore rigosità ai diversi fattori di qualità ed alla relazione tra qualità e mancanza d'errore, con esplicito riferimento ai costi.

Si vuole quindi rimarcare subito che il metodo dei FP è un metodo standard di quantificazione della dimensione funzionale, una base di riferimento su cui costruire ragionamenti di qualità, di difettosità e di costo di produzione.

Per applicazione s'intende una collezione coesa di procedure automatizzate, di relativi requisiti e dati che supportano un obiettivo di business. Ogni applicazione è separata dalle altre e dall'utente poiché è individuabile un confine che la contraddistingue. Il confine agisce come una "membrana" attraverso la quale passano i dati processati dalle transazioni di Input, Output e Inquiry. Il confine va visto da una prospettiva di business, di dominio logico e non va basato su considerazioni tecniche o fisiche. Anche la proprietà dei dati, l'utenza o una prevalenza di competenza di un ufficio rispetto ad un altro, possono essere di ausilio per meglio stabilire il confine di una applicazione.

La quantità di informazione racchiusa nel numero dei FP è garantita dal metodo utilizzato, dalla certificazione del personale che effettua il conteggio, dalla documentazione esistente. Essa è indipendente dalla tecnologia con cui l'applicazione è realizzata o utilizzata. Non è possibile parlare di FP senza far riferimento al metodo utilizzato, in quanto in passato sono stati definiti più metodi tra cui i più famosi sono MarkII, Boeing 3D, Feature Point, Nesma. Attualmente il metodo IFPUG 4.1.1 è utilizzato da circa il 90% degli utilizzatori di metodi dimensionali funzionali a livello mondiale, come evidenziabile dai dati dell'ISBSG. L'ISBSG ha raccolto circa 2.000 progetti relativi allo sviluppo ed evoluzione funzionale/normativa del software e si sta accingendo a raccogliere dati di progetto sulla manutenzione migliorativa, adeguativa (al mutare della tecnologia) e correttiva.

L'indipendenza dalla tecnologia dei metodi di dimensionamento funzionale è stata richiesta dallo standard ISO/IEC 14143 ed ha determinato progressivamente, nell'applicazione del metodo IFPUG, l'abbandono dei 14 fattori di aggiustamento insiti nella versione originaria del metodo, troppo legati a soluzioni tecnologiche (peraltro datate).

La standardizzazione che rende confrontabile la misura dei FP in termini funzionali in varie realtà applicative, e nel mondo, non deve trarre in inganno circa l'unicità del costo unitario che può differire invece a seconda di varie condizioni tecnologiche di sviluppo e delle leggi di mercato. Infatti il relativo costo unitario del FP è funzione di vari fattori, di cui se ne possono elencare alcuni a titolo esemplificativo: la criticità dell'ambiente e dell'applicazione, il linguaggio utilizzato, il processo metodologico seguito, la tecnologia di sviluppo e l'eventuale ausilio di sistemi autore generativi e l'utilizzo di package, l'esperienza del personale, la grandezza dei gruppi, la complessità dell'applicazione, la variabilità dei requisiti, il contesto d'uso, la possibilità di riuso, il coinvolgimento dell'utente nelle fasi produttive, la qualità del prodotto in tutti i suoi aspetti generali e la presenza di eventuali servizi indiretti offerti assieme al prodotto software. In una stessa organizzazione, specie se di grandi dimensioni, questi fattori possono risultare mediamente stabili o comunque compensarsi nel loro complesso conducendo all'uso di un costo unitario medio del FP, invece che l'adozione di vari costi differenziati caso per caso a seconda della tecnologia utilizzata.

La definizione rigorosa dei concetti relativi alla misurazione della dimensione funzionale del software (FSM) e la descrizione dei principi per applicare un metodo FSM sono contenuti nello standard ISO/IEC 14143-1 del 1997 "Information Technology - Software measurement - Functional size measurement - Definition of concepts". Gli International Standard emessi dal JTC1 (Joint Technical Committee), al termine di una serie di analisi ed approvazioni ufficiali, sono considerati tra i più autorevoli standard nel campo dell'Information Technology.

Un metodo FSM deve avere le seguenti caratteristiche:

- si basa su una rappresentazione dei requisiti utente vista da una prospettiva degli utenti;
- può essere applicato tempestivamente appena i requisiti funzionali utente sono stati definiti e sono disponibili;
- deriva la dimensione funzionale a partire da requisiti funzionali, indipendentemente dai requisiti tecnici e dalla qualità.

La dimensione funzionale è quindi indipendente dall'effort necessario allo sviluppo o alla manutenzione, dalle metodologie impiegate, dai supporti fisici utilizzati e dalle componenti tecnologiche. Viceversa l'effort, quindi il costo, dipenderà dai fattori citati. Un metodo FSM dovrebbe anche indicare il grado di convertibilità della dimensione con altri metodi dimensionali.

Grazie alla conformità allo standard ISO/IEC 14143 il metodo IFPUG 4.1.1 Unadjusted Function Point Method è stato approvato in ambito ISO ed è divenuto PAS (Publicly Available Specification) nel 2002.

La metrica dei FP si è sostituita gradualmente alle LOC (Line of Code: linee di codice in passato chiamate anche "schede", dal termine schede perforate che venivano lette da un particolare dispositivo hardware). Le LOC variano, a parità di funzioni svolte, secondo il tipo di linguaggio utilizzato e secondo gli stili di programmazione e la loro misurazione non è mai stata standardizzata a livello internazionale. Oggi sono più difficilmente quantificabili da quando si sono diffusi maggiormente linguaggi visuali avanzati per GUI e Web che, tra l'altro, prevedono possibilità di scrittura, a differenza del COBOL, non omogenee rispetto alla lunghezza fisica della linea stessa.

Secondo la Gartner Group i FP sono ancora la metrica più appropriata per misurare la dimensione di un'applicazione, presumibilmente almeno fino al 2005, anche se il numero delle imprese che adottano misure, ed in particolare i FP, sono solo il 25%. Tra le ragioni c'è il fatto che la metrica può sembrare difficile da applicare, specie in assenza di corsi di formazione e di personale qualificato.

Il conteggio dei FP applicato in modo canonico, cioè "forward" partendo dai requisiti, presuppone la conoscenza di regole precise ed è oggetto di nuove professioni per le quali sono richiesti appositi skill. Gli specialisti per il conteggio CFPS sono certificati dall'IFPUG anche attraverso la collaborazione del GUFPI/ISMA. Il metodo applicato invece empiricamente in modo "reverse" o con tecniche di "backfire", cioè partendo semplicemente dalle LOC, è in disuso e consisteva nel ricercare proporzioni tra i FP e le LOC; la sua attendibilità è relativa e da sperimentare eventualmente con appositi benchmarking nei propri ambienti.

In effetti i FP rappresentano qualcosa di più di una tecnica di conteggio in quanto contribuiscono ad un approfondimento delle funzionalità, producendo un miglioramento della analisi dei

requisiti, rendendone possibile una quantificazione e migliorando la documentazione del software. Essi costituiscono inoltre un'ottima opportunità per indirizzare verso veri e propri censimenti delle applicazioni esistenti nel sistema informativo.

Il metodo di conteggio, applicato con i dovuti criteri, si propone di rispondere alle seguenti caratteristiche:

- indipendenza dalla tecnologia e dalla qualità;
- ripetibilità della misura;
- sensibilità alla grandezza percepita dal produttore e dall'utente attraverso il numero di processi elementari e dei file logici con la relativa complessità;
- comparabilità metrica con altri metodi;
- accuratezza;
- apprendibilità (con circa 1 mese di teoria e 2-3 di pratica);
- usabilità ed efficienza (un calcolo di dimensionamento di una applicazione media richiede in genere da poche ore a 2-3 giorni);
- documentabilità;
- condivisibilità con l'utente e trasparenza;
- certificabilità del personale da adibire al conteggio
- manutenibilità e possibilità di supporto da parte di Associazioni e User Group.

Il metodo dei FP, come ogni sistema metrico, rientra in quelle attività di misurazione che ogni organizzazione complessa mette in atto per valutare la produzione ed orientare gli investimenti. Secondo Capers Jones circa il 2% della popolazione coinvolta nello sviluppo di software è dedicato alle metriche. Un tale investimento consente di disporre di un buon apparato metrico, non solo costituito dai FP, comprendente anche metriche di qualità e di servizio, fornendo peraltro dei benefici di risparmio complessivo.

2. Breve sintesi del metodo

I FP si concretizzano in una serie di punteggi (pesi) assegnati secondo le regole di conteggio a cinque rilevanti componenti o funzioni: Input (EI), Interrogazioni (EQ), Output (EO), File logici interni (ILF) ed File logici esterni all'applicazione (EIF); tali componenti sono evidenziabili dall'esame dell'applicazione e della sua documentazione.

Usando parole molto semplici si può dire che la tecnica fornisce una quantificazione delle informazioni, da un punto di vista logico, che entrano, escono e si memorizzano in un sistema attraverso l'esecuzione di una applicazione software. Si precisa che i termini "interno" e "esterno" presuppongono di aver determinato un limite, un confine che distingua l'applicazione da quantificare dalle altre applicazioni del sistema. Gli elementi oggetto di conteggio sono in relazione tra loro. In sintesi i File interni o esterni all'applicazione (ILF o EIF) sono primariamente e rispettivamente mantenuti o referenziati in diverso modo dalle attività elementari di Input (EI), Interrogazione (EQ), Output (EO); di seguito si riportano i loro intenti primari che le attività elementari intendono svolgere. Ad esempio un Input avrà la finalità principale di inserire dati in un file, ma secondariamente potrà anche leggere informazioni su un altro file (tra parentesi sono indicati gli intenti non primari):

Attività elementari	File logico interno	File logico esterno
	ILF	EIF
Input esterni (EI)	scrittura (lettura)	(lettura)
Interrogazioni esterne (EQ)	presentare	presentare
Output esterni (EO)	presentare (scrittura)	presentare

Un esempio di processo di Input (EI) si verifica con l'acquire dei dati e memorizzarli in un file interno all'applicazione, un esempio di Interrogazione (EQ) con il fornire una semplice risposta ad una domanda, prelevata da un file interno o esterno, ed un esempio di Output (EO) con lo stampare dei dati letti da un file interno o esterno e su cui sono effettuati calcoli; queste attività costituiscono processi elementari, cioè le più piccole unità di attività significative per l'utente

Alcuni verbi che identificano i processi elementari per un EI, e connessi in genere con uno o più ILF, sono: acquisire, inserire, aggiungere, inviare dati al sistema, impostare, importare, popolare, assegnare, pianificare, schedulare, cancellare, eliminare, modificare, aggiornare, alterare, accettare, variare, rivedere, controllare, validare, memorizzare, mantenere

Alcuni verbi che identificano un EO, e che riferenziano ILF o EIF, sono: calcolare e contabilizzare, oltre a: fornire (*), stampare (*), visualizzare (*), produrre (*), trasmettere (*), esportare (*), inviare dati da parte del computer (*), presentare (*), pubblicare (*), i quali possono essere anche EQ se non contengono calcoli. Altri verbi che identificano un EQ, se non ci sono dati derivati, e che riferenziano ILF o EIF, sono (oltre quelli precedenti già citati contrassegnati da *): interrogare, ricercare, decodificare, leggere, accedere, listare.

Di seguito si riepilogano i Function Point (cioè i punteggi funzionali) da attribuire alle funzioni secondo la loro complessità bassa, media, alta (che dipende, ma non se ne riporta il dettaglio, dal numero di dati elementari trattati, da come sono raggruppati, da quanti file riferenziano):

Funzioni	Complessità		
	bassa	media	alta
Input esterni (EI)	3	4	6
Interrogazioni esterne (EQ)	3	4	6
Output esterni (EO)	4	5	7
File esterni di interfaccia (EIF)	5	7	10
File interni logici (ILF)	7	10	15

Tali tabelle di pesi convenzionali sono estratte dal Manuale sulla pratica di conteggio © IFPUG e sono riconosciute ed usate a livello internazionale, consentendo comparazioni anche tra i diversi paesi in cui attualmente il metodo è presente, anche con l'assistenza di Associazioni specifiche (Europa, Stati Uniti, Brasile, India, Corea, Australia, Giappone). I criteri per determinare la complessità, come tutti i punteggi citati, sono riportati nel suddetto manuale (CPM 4.1.1). Attraverso tali criteri si perviene, per ognuno degli elementi oggetto di conteggio, a stabilire la loro complessità, sulla base di tabelle, in funzione

del numero dei dati trattati non ripetuti riconosciuti dall'utente, dei sottogruppi logici all'interno di un file logico, del numero dei file logici letti o aggiornati.

Si possono infine sintetizzare le seguenti regole logiche di processo:

se sono effettuate validazioni, conversioni, selezioni, condizioni, sort, si può essere in presenza di EI, EO, EQ;

- se sono effettuati calcoli matematici, o sono creati dati derivati, si può essere in presenza di EI, sicuramente di EO, non di EQ;
- se si effettua almeno un aggiornamento di File, o si altera il funzionamento del sistema, si è in presenza di almeno un EI o EO, non di EQ;
- se si riferenzia un ILF o EIF, o se si elaborano dati o informazioni di controllo, si può essere in presenza di EI, EO, sicuramente di EQ;
- se si presentano informazioni al di fuori del confine dell'applicazione si può essere in presenza di EI, ma sicuramente di EO o EQ;
- se si acquisiscono dati o informazioni di controllo all'interno del confine dell'applicazione si è in presenza di un EI e si può essere in presenza di un EO o EQ.

Il numero di FP è quindi proporzionale alle funzione da automatizzare: acquisire dati, produrre report, aggiornare archivi, ecc. Il numero potrebbe rimanere di per sé costante per l'intera applicazione, ma generalmente risulta in media in lieve crescita nel corso del ciclo di vita del software, rispetto al momento iniziale della stima, anche se non si può escludere a priori una sua diminuzione. Nelle fasi iniziali di un progetto infatti i requisiti utente evolvono rapidamente e diventano via via più precisi e dettagliati. Tale fenomeno è conosciuto e talvolta indicato col termine "obiettivi striscianti" (scope creep), come citato anche nel manuale IFPUG 4.1.1 "Manuale regole di conteggio" del 2000, pag. 4-3. Il continuo confronto tra l'utente e chi sviluppa porta ad un'evoluzione dei requisiti ed è necessario che vi sia una intesa tra l'utente e chi sviluppa, su quali siano le funzionalità che devono essere incluse nell'applicazione per raggiungere un grado di consistenza e completezza dei requisiti funzionali da soddisfare.

Prima di iniziare un conteggio dei FP, è sempre necessario determinare la fase del ciclo di vita nello sviluppo del software. E' importante tenere presente che i conteggi iniziali di FP sono delle stime delle funzionalità che verranno rilasciate. Non appena risultano chiari gli obiettivi, è piuttosto normale identificare delle ulteriori funzionalità che non erano state specificate nei requisiti iniziali. E' essenziale aggiornare il conteggio iniziale al termine del progetto; il numero di FP finale deve misurare accuratamente tutte le funzionalità rilasciate all'utente.

Risultano di conseguenza vari documenti da gestire da parte dei diversi responsabili delle fasi del processo per tenere traccia delle evoluzioni. Il processo di conteggio dei FP può essere espresso da un workflow che descrive le varie fasi di conteggio e mette in connessione eventuali diverse applicazioni e tool di supporto, di mercato o realizzati ad hoc, che si limitano al momen-

to a supportare la conservazione dei dati e dei documenti e ad effettuare stime. Viceversa il conteggio vero e proprio è un'attività "umana" non ancora automatizzata (anche se sono in corso vari tentativi con diverse tecniche) che in genere contribuisce ad agevolare la comunicazione sui requisiti tra committente/utente e fornitore.

Vediamo nel seguente paragrafo una possibile raffigurazione di tale processo, utilizzando Use Case Diagram e Activity Diagram UML (Unified Modeling Language).

3. Workflow sui Function Point

Un workflow rappresenta un flusso di lavoro automatizzabile. Quando più attori con diversi ruoli interagiscono nel tempo scambiandosi documenti esso risulta particolarmente efficace ed utile potendo collegare azioni, anche concorrenti, che si sviluppano nel tempo utilizzando a volte strumenti diversi.

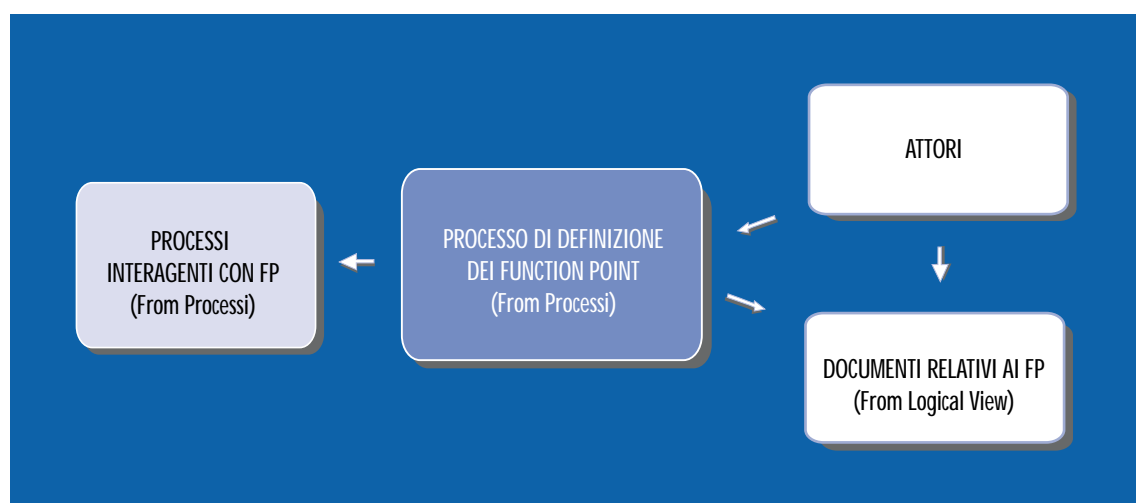
Il processo di conteggio dei FP, relativo al processo di sviluppo del software, è connesso con altri processi interagenti, come ad esempio: la pianificazione, la manutenzione, la fatturazione.

Gli attori coinvolti nel processo di conteggio hanno diversi ruoli e task specifici.

Ad esempio alcuni attori possono avere il ruolo di effettuare la stima (ed in genere conoscono sia la tecnica di conteggio che il dominio applicativo) ed altri di approvare il volume complessivo dei FP in relazione alle disponibilità di spesa ed ai requisiti richiesti.

Infine i documenti che tracciano i requisiti, i conteggi e le comunicazioni tra le parti sono un numero non irrilevante e necessitano anch'essi di essere gestiti e monitorati con strumenti automatici. In sintesi si può raffigurare con il seguente schema l'evidenziazione di interconnessione tra processi, attori e documenti:

Figura 8: Schema di interconnessione tra processi, attori, documenti

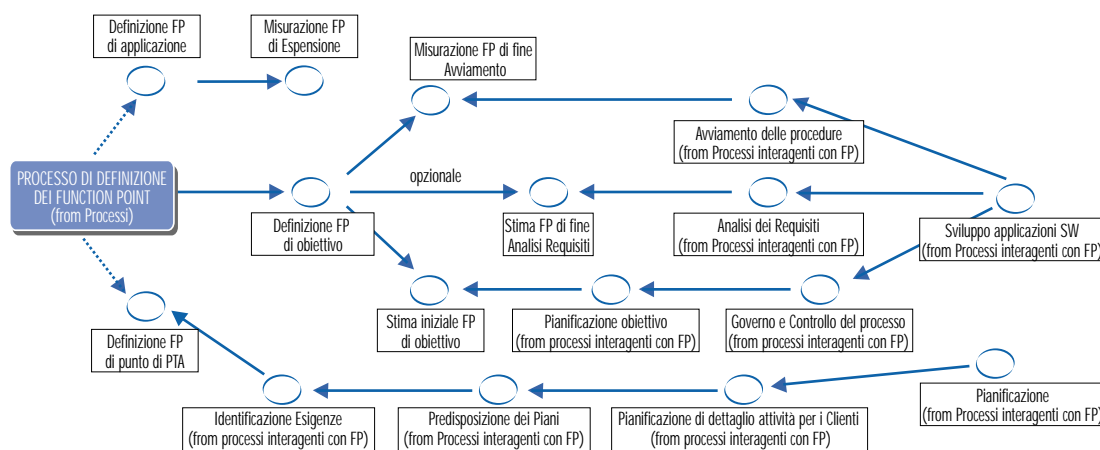


Il processo centrale di definizione dei FP è a sua volta scomponibile in diverse fasi tra cui citiamo quelle essenziali per il processo di contabilizzazione in un obiettivo di sviluppo:

- Definizione dei FP secondo i requisiti dell'obiettivo da automatizzare espressi nei Piani Tecnici secondo l'identificazione delle esigenze. Lo scopo principale di questa fase è quello di descrivere l'obiettivo e stabilirne i confini. Ciò può comprendere in particolare l'individuazione di tutte le entità esterne che interagiscono con il sistema (attori), la descrizione dei primi macro use-case, una valutazione iniziale dei rischi e una stima delle risorse necessarie.
- Stima iniziale del numero dei FP a livello di obiettivo conseguente alle attività suddette
- Successiva stima dei FP a fine analisi dei requisiti conseguente allo sviluppo di un piano dettagliato dell'obiettivo e analisi del problema.
- Conferma della misurazione dei FP a fine avviamento che tenga conto di eventuali variazioni in corso d'opera.

Nel caso di un obiettivo evolutivo su una applicazione pre-esistente seguirà una attività di misurazione dell'intera applicazione risultante.

Figura 9: Processo di definizione dei Function Point



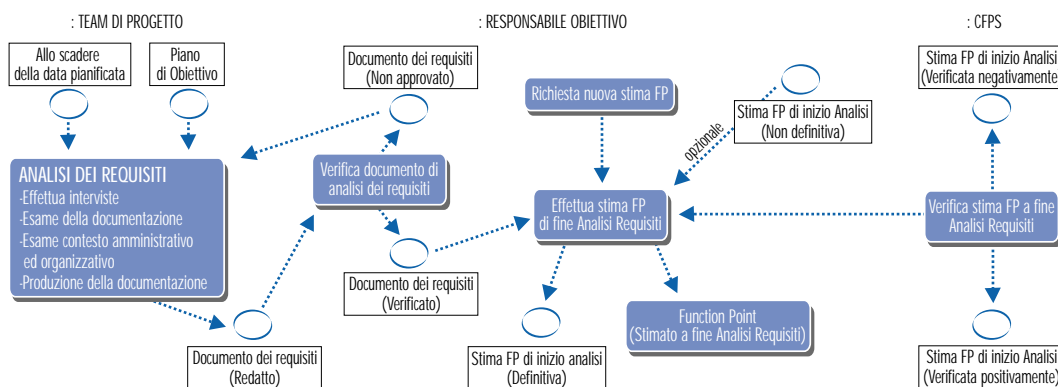
Le attività citate evidenziano una stretta connessione tra processi di pianificazione e attività di stima e misurazione proprie dello sviluppo delle applicazioni software. Infatti il processo di definizione dei FP è strettamente collegato al processo di sviluppo del software ed alla fase di previsione degli impegni economici e pianificazione operativa da effettuare in sede di compilazione del piano delle attività di sviluppo, manutenzione e conduzione del sistema informativo per il committente. Il flusso di previsione e consuntivazione dei Fp è inoltre interagente sia con i processi aziendali necessari a monitorare il perseguimento degli obiettivi economici e lo stato di avanzamento delle attività che con il processo di emissione delle fatture per i corrispettivi dovuti.

Si riporta infine, ad esempio, un Activity Diagram, limitatamente alla fase di Analisi dei Requisiti, nella quale intervengono per la stima dei FP vari attori:

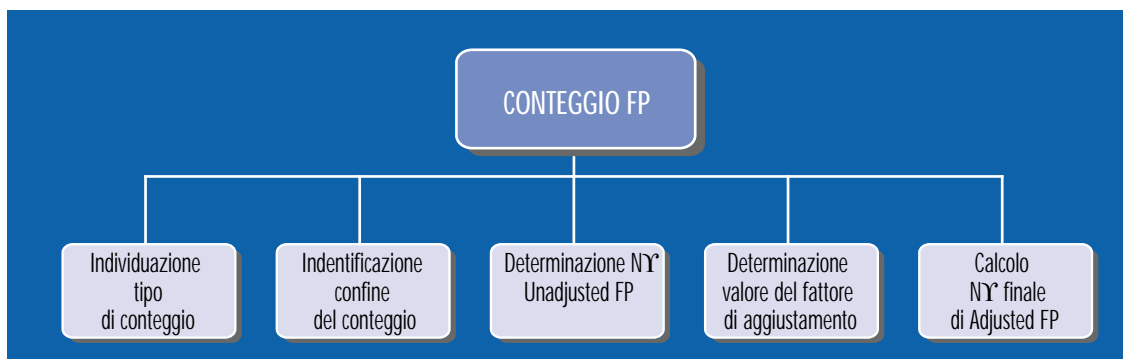
- team di progetto;
- responsabile obiettivo;
- CFPS.

Il team di progetto richiede ai vari responsabili obiettivo di descrivere gli obiettivi di cui effettuare una stima secondo le esigenze utente; i responsabili si avvalgono all'occorrenza della consulenza di persone certificate CFPS. Sono evidenziati i vari documenti utilizzati per documentare il processo, come il piano dell'obiettivo, i documenti sui requisiti, la stima di FP ecc..

Figura 10: Activity Diagram: analisi dei requisiti



Il vantaggio di usare workflow descrittivi del processo, oltre a illustrare in sintesi i passi da svolgere, è quello di fornire all'istante i facsimile dei moduli necessari per la documentazione, ma ad esempio anche fornire funzioni di assistenza per le varie fasi, tipo help in linea, evidenziando i passi del manuale di conteggio specifici per assistere gli specialisti di conteggio nei loro calcoli. Inoltre un workflow tendente all'automazione del processo può attivare al momento giusto le singole applicazioni di archiviazione dati e/o visualizzazione report, come infine assistere gli attori, secondo i loro ruoli e competenze, a tracciare lo stato di avanzamento e ad eseguire in ordine la sequenza di tutti i task, come in particolare:



4. Applicabilità del metodo dimensionale

Il metodo nella sua generalità può essere applicato a fini dimensionali a qualsiasi obiettivo o applicazione, sia di tipo tradizionale che innovativo, che abbia un confine e che riceve, o invia, dati attraverso questo confine.

La necessaria elencazione dei processi elementari conduce ad una suddivisione del sistema in piccole componenti che possono essere meglio capite ed analizzate. Se tale scomposizione è effettuata nelle fasi di analisi, quando possibile, essa dovrebbe non rappresentare un aggravio eccessivo, specialmente per gli ambienti di produzione tradizionali.

Negli ambienti innovativi (intesi qui ad esempio come il mondo di alcuni Web e di alcuni Data Warehouse) la sua applicabilità, anche solo a fini dimensionali, molto dipende dalla fase del ciclo di vita, dalla documentazione utilizzata, dai metodi di produzione, dagli strumenti di supporto utilizzati, dall'eventuale utilizzo del prototyping o della JAD (Joint Application Design). L'eventuale instabilità dei requisiti riscontrabili in questi metodi potrebbe consigliare di effettuare conteggi successivamente, essendo oneroso il conteggio in corso d'opera, superando qualche giornata media di impegno, ritenuta accettabile nei casi di requisiti stabili.

Quanto segue è riportato a titolo esemplificativo per illustrare come certi oggetti utilizzati in ambienti innovativi possono essere, in certe condizioni, identificati e trattati dal metodo dei FP, in quanto le linee guida di conteggio ufficiali sulle nuove tecnologie sono ancora in via di pubblicazione da parte dell'IFPUG e costituiranno presto un riferimento e completamento per alcuni aspetti che il CPM 4.1.1 non precisava.

Un sito web può contenere una o più applicazioni o servizi e in tal caso si chiama in genere portale. In ogni caso è in genere evidente una delimitazione di contenuto, oltre che funzionale, tra un sito web ed un altro o tra un'applicazione ed un'altra.

I siti web possono distinguersi in informativi (retrieval o publishing) e interattivi (transazionali o telematici), anche se la differenza a volte non è marcata ed è possibile effettuarla solo con criteri di prevalenza. I siti interattivi possono essere paragonati in alcuni casi ad applicazioni tradizionali che prima erano realizzate con applicazioni on line, su reti dedicate oppure su sistemi client/server, e che ora risiedono su internet. I siti informativi, specie se multimediali, e comunque parti dei siti interattivi, possono presentare delle particolarità non gestite dal conteggio dei FP. Nei siti informativi di frequente si evidenziano solo EQ, EO e EIF, in quanto non viene aggiornato alcun file (se non dal webmaster o webengineer con una differente applicazione di quella vista dall'utente finale e qui presa in esame). Nei siti interattivi si evidenziano invece comunemente, per definizione, anche EI e ILF, attivati dall'utente finale.

In genere si può contare a parte il conteggio per la manutenzione del contenuto del sito da parte di un webengineer (con ulteriori EI e ILF), spesso servita da strumenti di web content, e non bisogna inoltre confondere l'eventuale lavoro redazionale per l'aggiornamento del sito, il reperimento delle informazioni o attività di data-entry, che nulla hanno a che fare con il conteggio dei FP di sviluppo, ma semmai con l'onere di conduzione del sito

Secondo opinioni attualmente prevalenti non sono comunque oggetto di conteggio i bottoni/barre di navigazione, né i link di ipertesto. Così come non si contano i contatori di accesso, né le search con motori di ricerca sviluppati e gestiti al di fuori della applicazione in esame, né le e-mail che attivano semplicemente un programma di posta elettronica.

Prescindendo dal processo utilizzato e dagli strumenti impegnati si riporta ad esempio una breve descrizione di un possibile semplice conteggio di sito web di tipo publishing, cioè senza interazioni che acquisiscono informazioni dall'utente. Si ipotizza un sito web che, visto da un utente finale, tratta di sei argomenti logici diversi che richiedono competenze diverse sia per la loro definizione che per la loro comprensione.

Ogni argomento logico di sito web di tipo publishing risulta formato da alcune pagine logiche o File Logici indipendenti identificabili dall'utente attraverso l'utilizzo del sito. Poiché tali file logici non sono mantenibili (modificabili) dall'utente, se non dal webmaster con un'applicazione di conteggio, come accennato, da valutarsi separatamente, essi sono equiparabili ad EIF, eventualmente scomposti in sottoargomenti interrogabili identificando RET e DET.

Si individuano parallelamente le funzioni di interrogazione EQ, ciascuna che insiste su ogni File con un tasto e relativo testo per le visualizzazioni delle relative pagine logiche, identificando FTR e DET

Si può ipotizzare che non si individuano EO in quanto non sono previste visualizzazioni che comportano dati calcolati o derivati.

Essendo un sito publishing non si individuano EI in quanto non sono previste immissioni di dati, questionari, guest book, bulletin board, aggiornamenti, definizioni di password, personalizzazioni, mailing list, contatori, ecc. Non si individuano neanche ILF in quanto l'utente finale non manutene direttamente alcun file.

Anche per la quantificazione della dimensione funzionale dei Data Warehouse possono valere analoghe regole di conteggio. E' possibile infatti stabilire una corrispondenza tra concetti della misura funzionale e le caratteristiche dei sistemi Data Warehouse.

A tal fine occorre distinguere l'applicazione di popolamento (assistite in genere da software ETL – Extract Transformation Loading) da quella di interrogazione vera e propria (assistita da software OLAP – On-line Analytical Processing). Tale ultima parte è assimilabile ad un sito web interattivo. E' utile ricordare che l'approccio metodologico prevalentemente scelto per lo sviluppo di progetti di tipo Data Warehouse prevede l'implementazione di un Data Warehouse multi-tier, comprendente un EDW (Enterprise Data Warehouse) e vari Data Mart corrispondenti a viste specifiche.

Per l'applicazione di popolamento e aggiornamento, le funzioni ILF si identificano nei file interni creati di tipo "stella logica" o "Data Mart" che racchiudono i gruppi logici di dati riconoscibile dall'utente. Le funzioni di EI si identificano per ogni catena distinta ed autonoma di passi elaborativi di caricamento e aggiornamento che conduce alla scrittura unitaria di uno o più ILF, i relativi RET e DET possono essere connessi rispettivamente con le "fact table", laddove queste non siano invece considerate ILF, e con gli attributi logici calcolati o reperiti.

Si identificano eventuali EIF in corrispondenza di ogni archivio logico di un connesso sistema operativo necessario in lettura alla fase di caricamento e aggiornamento.

Tale processo di popolamento rischia un notevole sottodimensionamento della produttività in FP nel caso di complessità nel reperimento delle fonti, nella gestione dell'input con estrema necessità di attivazione di procedure di "data cleaning" e processi di trasformazione.

Per l'applicazione relativa alle interrogazioni on line, supponendo che l'utente non possa modificare alcun dato, si identifica come EIF ogni archivio logico indipendente costruito dall'applicazione di popolamento (in quest'ultimo ambito era stato considerato ILF) o, in alternativa, ogni file logico di un connesso sistema operativo necessario in lettura alla fase di caricamento o aggiornamento del Data Mart, non contando i duplicati fisici su aree/piattaforme differenti. Si può identificare un solo EO o un solo EQ per ogni file logico identificato; non si identificano ulteriori EO/EQ per viste di dati che si differenziano per il solo uso di un differente livello di una gerarchia di sintesi sui dati del file logico di partenza, mentre si possono identificare ulteriori EO/EQ, per una medesima "stella logica" o "Data Mart" o file logico, nel caso di viste, con dati elementari differenti, sul file logico di partenza, esplicitamente richiesti dall'utente.

L'applicabilità del metodo di dimensionamento funzionale è, in conclusione, generale, ma devono valere nella pratica considerazioni di efficienza e complessità del conteggio. Per gli ambienti innovativi, l'uso del conteggio in FP è spesso limitato a fornire un riferimento di base per porre la difettosità in relazione al volume delle funzioni sviluppate od offerte dall'applicazione in esercizio, mentre è difficilmente utilizzabile per finalità economiche a causa di notevole ricorso a strumenti automatici di supporto di tipo generativo o parametrico.

5. Limitazioni nell'utilizzo economico

Se da un lato il metodo dei FP è applicabile in vari ambienti per le sue finalità dimensionali, esso può presentare possibilità di applicazione nell'uso economico solo in determinate condizioni. Si può considerare che negli ambienti innovativi le capacità generative dei mezzi di produzione sono tali che il costo complessivo del prodotto risente di meno della quota di personalizzazione collegata alla dimensione, e quindi all'effort, facendo prevalere invece il costo di licenza dei tool generativi che possono divenire preponderanti. In tali casi mentre sui fattori dimensionali giocano un grande ruolo le rigorosità delle regole di applicazione, sui fattori economici intervengono regole produttive e di mercato generando differenziazioni di costo.

Questo aspetto di limitazione riguarda tutti gli ambienti innovativi, compresi alcuni Web o alcuni Data Warehouse, laddove il massiccio riuso strutturale di funzioni di costruzione e di interrogazione sono molto agevolate da prodotti automatici (chiamati anche Case – Computer Aided Software Engineering). In particolare, ad esempio, un caso estremo si ritrova nell'ambiente ERP dove i FP contabilizzati per la parte di "personalizzazione/evoluzione" non è certamente esaustiva rispetto alle funzioni ottenute con "parametrazioni" di quelle fornite dal sistema ed a volte non è neanche facile farne una distinzione. Inoltre le funzioni "automatiche" di cui si è "venuti in possesso" non sono stabilmente correlate con un effort aggiuntivo tradizionale, ma con il valore una licenza d'uso che consente facili parametrizzazioni.

Ogni volta che i FP “realizzati ad hoc” non costituiscono la parte significativa dei FP “consegnati” il metodo dei FP, seppur applicabile in linea teorica per motivi dimensionali, non è utile per la determinazione del costo di tutta la produzione, in quanto la parte di FP connessi ad attività “manuali” di personalizzazione è minoritaria rispetto alla parte generata

Tali limitazioni spingono i ricercatori verso la determinazione di nuove misure per la stima dei costi o di determinazione di diversi costi unitari per FP. In certi sistemi software, come quelli CRM, si ravvede la necessità di diverse e nuove misure e di modelli specifici, che oltrepassino gli ambienti dimensionali funzionali caratteristici dell'applicazione software e quantifichino anche livelli di servizio, tipologie varie di attività dirette e indirette, livelli di utenza.

Un diverso tipo di limitazione, che può combinarsi al precedente, può risultare per quel tipo di applicazioni la cui misura funzionale non viene più percepita dal metodo di conteggio a causa dell'uso di attribuzioni di pesi con tabelle aperte, come nel caso di alcune procedure di popolamento di Data Warehouse, spesso automatizzate da soluzioni ETL. In tal caso le misurazioni effettuate, come accennato, risultano sovente errate per difetto ed i FP contabilizzati meno di quelli che si otterrebbero con tabelle chiuse con numero elevato di intervalli progressivi. In effetti in tutti i metodi che operano con tabelle aperte nell'attribuzione dei pesi non sempre si riesce, oltre un certo livello di complessità, a fotografare la vera ampiezza dimensionale dell'applicazione. In altre parole può verificarsi il caso che pur aggiungendosi molta dimensione funzionale, e quindi effort, il metodo non riesce a tracciarla.

In quest'ultimo caso le ricerche sono attive per migliorare la sensibilità e l'accuratezza della stessa misura dimensionale.

In conclusione le limitazioni derivano da un combinato di fattori: da una parte una incapacità dei metodi di conteggio con tabelle aperte di essere sensibili al crescere della complessità, fornendo quindi misure dimensionali per difetto; dall'altra la necessità di un forte ricorso a strumenti automatici di supporto che riducono la correlazione tra costo e dimensione, generando ricerche o verso nuovi modelli di costo o verso una forte differenziazione dei costi unitari del FP secondo modelli tassonomici di produzione.

Le direzioni di ricerca vanno quindi da un lato verso misure di dimensioni funzionali, sempre, si auspica, rapide nel conteggio, ma che siano sensibili, in un maggior numero di casi, alla grandezza dell'applicazione. Recentemente nel 1998 il consorzio COSMIC (Common Software Measurement International Consortium) ha proposto ad esempio una nuova misura, ancora sperimentale anche se PAS-ISO, chiamata COSMIC FFP utile per la dimensione funzionale del software e che risente meno di questa limitazione sulla progressività della grandezza. Sia il metodo IF-PUG che COSMIC riconoscono comunque importanti per la dimensione funzionale:

- l'importanza dei processi elementari come unità da misurare;
- la movimentazione dei dati trattati da un processo;
- gli accessi ai dati

mentre nessuno dei due misura specificatamente la complessità degli algoritmi, calcoli, ecc..

In altre direzioni si effettuano ricerche su nuovi modelli di costo che tengano conto delle po-

tenzialità generative di software di supporto e su costi differenziati.

In ogni caso rimuovendo tali limitazioni si dovrà comunque tener conto nell'introduzione di nuovi metodi di alcuni aspetti tra cui:

- necessità di aggiornamenti formativi;
- attesa di maturazione delle Organizzazioni associative per l'assistenza metodologica e la certificazione;
- necessità di raccolta di valori unitari di costo e dati storici per valutare congruità;
- sperimentazione della efficienza di conteggio (e durata in giorni) perché nei casi di abolizione di tabelle aperte non è possibile interrompere il conteggio per superamento dei limiti ed occorre portarlo a termine con un aggravio di tempo da verificare.

6. Indicatori di valutazione

L'uso per valutazioni qualitative, più che economiche, rimane comunque un aspetto interessante che merita un approfondimento ed altre ricerche.

Ad esempio i difetti riscontrati in esercizio possono essere proporzionati sulla base della dimensione del sistema:

Defect density = Difetti / Function Point

Un altro indicatore si può ottenere moltiplicando i FP per il numero degli utenti generando un "fattore d'uso" che rispecchia la potenzialità d'uso delle funzioni con la quale sono correlate vari ulteriori fenomeni come, ad esempio, la penetrazione sul mercato:

Usage FP = Function Point on line * Utenti

disponendo di ulteriori FP da realizzare o Utenti da raggiungere è possibile costruire indicatori di penetrazione e impatto.

L'indicatore di "leverage" per valutare l'effettiva efficacia delle applicazioni può essere costruito rapportando, ad esempio, le richieste informative da Call Center ai FP effettivamente in esercizio. Una possibile quantificazione, che tenga conto anche della frequenza di utilizzo, consentendo confronti tra situazioni diverse, è la seguente:

Running FP = Function Point on line * Frequenza standardizzata * Utenti

Ove:

- Running FP: i FP effettivamente in uso
- Function Point on line: i FP in linea che danno servizio (sono esclusi quindi i FP di procedure batch di ausilio)
- Frequenza standardizzata: la frequenza media del periodo considerato è uguale a 1 per usi giornalieri, a 0,5 per usi bi-giornalieri ecc. (la frequenza standard è pari a 1/giorni di intervallo medio tra un uso ed il successivo; ad esempio una frequenza annuale è pari a $1/365 = 0,003$)
- Utenti: numero medio di utenti del periodo considerato

Ad esempio, un servizio di 300 FP che è eseguito ogni giorno da 1.000 utenti ha un Running FP pari a 300.000 ($300 * 1 * 1.000$), mentre un altro servizio, sempre di 300 FP, che è eseguito settimanalmente da 2.000 utenti ha un Running FP pari a 84.000 ($300 * 0,14 * 2.000$).

Se si volesse effettuare un'analisi delle chiamate di un Call Center correlate all'uso di software,

nell'ipotesi che il primo servizio avesse generato 5.000 chiamate/giorno al Call center e il secondo 4.000/settimana (trasformabile in giorni), avremmo per il primo un'incidenza percentuale dell'1,7 (5.000/300.000) e per il secondo del 0,7 (4.000/7/84.000).

Tali indicatori, con eventuali ponderazioni e aggiustamenti, possono essere applicabili allo studio di diversi fenomeni, suggerire indizi e condurre a diverse analisi le cui conclusioni sono da verificare caso per caso, tenendo anche presente fattori legati alla conoscenza dell'utenza del servizio connessa al tempo di esercizio.

7. Aspetti organizzativi

Ci sono delle condizioni favorevoli all'uso dei FP, basato su metodi che devono essere esplicitamente riportati su un manuale ufficiale di conteggio, come il Counting Practice Manual CPM 4.1.1. dell'IFPUG.

Lo skill delle persone che effettuano il conteggio non è un elemento da sottovalutare. Le competenze da possedere sono, in sintesi, derivanti dalle seguenti condizioni:

- aver frequentato specifici corsi di formazione;
- conoscere a perfezione il manuale di conteggio;
- aver fatto pratica di conteggio;
- aver superato l'esame di certificazione CFPS dell'IFPUG.

Tale certificazione è di tipo personale, non aziendale, riconosce ai CFPS la capacità di un conteggio corretto. A partire dal 2004-2005 esisteranno anche ulteriori certificazioni IFPUG:

- CFPS Advanced, che si baserà esclusivamente sull'applicazione dei FP alle nuove tecnologie;
- METRICS, che riguarderà la conoscenza delle problematiche sulle metriche del software valutabile in cinque categorie: esperienza, formazione, contributi professionali, pubblicazioni e insegnamento.

In particolare è auspicabile che la certificazione IFPUG sia ottenuta non solo da persone del fornitore ma anche del committente, al fine di favorire la comunicazione e le verifiche.

La ridottissima, o nulla, variabilità dei risultati di conteggio condotti da persone differenti, dipende principalmente da questa preparazione e certificazione.

Gli esperti di conteggio sono inoltre persone che in genere sanno dialogare abilmente, con la loro competenza tecnica di conteggio, con il collega esperto del dominio dell'applicazione, si pongono con atteggiamenti collaborativi, di mediazione e di indagine conoscitiva, creano soluzioni nuove ed amano approfondire i problemi in modo dettagliato. Hanno uno spirito di organizzazione e sanno intervistare le persone, oltre che comprendere la documentazione scritta. Sanno insegnare quello che hanno appreso. Hanno un senso di responsabilità e possono essere un utile riferimento per i responsabili della pianificazione dei progetti ed i coordinatori della raccolta documentale.

L'applicazione di metriche del software richiede una fase di crescita culturale che va favorita anche con seminari formativi che descrivono le finalità delle metriche e l'uso che se ne intende fare. Essendo al momento la tecnica dei FP non automatizzabile, ma svolta da esperti, è importante la massima trasparenza e comunicazione, oltre che di documentazione, tra l'esperto di dominio (che po-

trebbe non conoscere la tecnica) e l'esperto di conteggio (che potrebbe non conoscere la materia). Un impatto organizzativo indiretto, che si verifica all'avviarsi con l'uso della tecnica dei FP su larga scala, è quello derivante dal fatto che le persone che effettuano i conteggi vengono a conoscersi e frequentarsi, anche per scambiarsi esperienze e prassi. Praticamente si costituiscono certi tipi di "comunità di pratica", termine utilizzato spesso nei piani di Knowledge Management, in quanto i gruppi di esperti, anche informali, privilegiano lo scambio di informazioni su un argomento di comune interesse.

Un beneficio indiretto della tecnica dei FP, che ha anche risvolti organizzativi, è dato anche dalla maggiore conoscenza delle applicazioni, cioè una sorta di "portafoglio prodotti" del sistema, che conduce ad una maggiore conoscenza anche delle organizzazioni che gravitano attorno alle applicazioni.

Affinché un programma di metriche abbia successo occorre infine prevedere una serie di azioni, in breve così sintetizzabili:

- definire gli obiettivi di misurazione ed un piano di attuazione;
- ottenere la sponsorizzazione del Senior Management;
- assegnare al programma risorse dedicate;
- effettuare attività formative ed esercitazioni;
- considerare le misure come integrate nei processi;
- focalizzarsi sui risultati dei progetti;
- non misurare singoli individui;
- definire report di feedback;
- automatizzare il più possibile la raccolta;
- saper gestire la gradualità dell'introduzione di metriche e della disponibilità di risultati, alcuni dei quali saranno disponibili in tempi brevi ed altri nel lungo periodo.

Al fine di garantire il massimo aggiornamento del personale, è anche utile che le persone esperte in metriche partecipino ad associazioni di settore, convegni specialistici, working group.

8. Conclusioni

Il metodo dei FP è orientato alla quantificazione della dimensione funzionale. La dimensione funzionale è calcolabile in linea teorica per ogni tipo di applicazione, tradizionale ed innovativa, ed in parte anche per altri ambienti. Ragioni di efficienza nel conteggio, di applicabilità per scopi economici, imputabile al maggior uso di software di supporto o carenza di valori di produttività di riferimento, ne può suggerire un minor o maggior impiego.

E' infatti sempre più rilevante nelle più avanzate tecniche di produzione degli ambienti innovativi l'uso di strumenti di produzione parametrizzabili e generativi, riducendo la correlazione tra dimensione finale dell'applicazione complessiva realizzata e l'effort impiegato (costo) ed introducendo in modo rilevante l'incidenza del costo di licenze per l'uso di software parametrizzabile.

In sintesi lo scenario risultante in media, ottenuto anche sulla base delle esperienze conosciute

in ambito dell'Associazione Italiana sui FP, fornisce la seguente situazione di possibili usi dei FP:

Possibili usi del FP

Descrizione Ambienti	Usi		
	Dimensione funzionale	Costo unitario	Indicatori di valutazione
Ambienti tradizionali, Web interattivi, DW On line			
- Sviluppo/evoluzione	diffusa	differenziato	-
- Manutenzione	diffusa	valore medio	sì, difettosità
Ambienti innovativi: Web publishing/multimediali, DW ETL:			
- Sviluppo/evoluzione	diffusione condizionata	differenziato	-
- Manutenzione	diffusione condizionata	valore medio-raro	possibile, difettosità
ERP, CRM			
- Sviluppo/evoluzione	sì, su personalizzazione	differenziato	-
	no, su parametrizzazione	modelli ad hoc	modelli ad hoc
- Manutenzione	sì, su personalizzazione	valore medio-raro	-
	no, su parametrizzazione	modelli ad hoc	modelli ad hoc

ove per modelli ad hoc si intendono stime di costo in studio che combinano effort, prodotti e licenze d'uso.

La ricerca di altre metriche è in atto, in varie sedi, come accennato anche in ambito ISO, IFPUG, ISBSG e COSMIC, anche con l'orientamento di allargamento della visuale che va dall'analisi della quantità interna funzionale e dimensionale, verso altre caratteristiche di interesse per l'utente finale e riscontrabili nella qualità in uso del software.

Con riferimento allo standard ISO/IEC 9126 del 2000, che tratta dei fattori di qualità interni (statici) ed esterni (test utente) del prodotto software, si potranno inserire anche altri fattori nei modelli di riferimento che tengano conto di:

- funzionalità (misurabile in certe condizioni ancora dai FP);
- manutenibilità;
- affidabilità;
- efficienza;
- portabilità;
- usabilità;

mirando alla qualità in uso, cioè alla:

- produttività utente;
- gradevolezza del prodotto;
- sicurezza indotta;
- customer satisfaction.

Man mano che il software diventerà sempre più simbolico ed usabile, ad alto livello, anche le metriche si eleveranno in astrazione, come è già stato nel passaggio dalle LOC ai FP. Tale tendenza terrà sempre conto di ciò che l'utente percepisce, valore sempre valorizzato dalla tecnica dei FP e che ha segnato un importante salto di qualità nel campo delle metriche.

Anche introducendo altre metriche sarà essenziale continuare ad avere, nel tempo, un metro di

paragone definito, una unità di misura confrontabile e un metodo standard riconosciuto in ambito internazionale. Il trend è di migliorare la capacità di distinguere tra le differenti problematiche degli ambienti tradizionali o innovativi, con utilizzo specifico di nuovi mezzi di produzione, di tener distinti i processi di sviluppo e manutenzione, di perfezionare le misure quantitative (dimensionali) e quelle qualitative, per una migliore previsione e consuntivazione della qualità percepita e dei costi di produzione .

Bibliografia

- J. Brian Dreger, Function Point Analysis, Prentice-Hall, 1989
D. Herron, Function Point Analysis, Technology Transfer, Roma 1-2 Aprile 1992
C. Jones, Applied Software Measurement, Technology Transfer, Roma 10-11 Maggio 1993
A. Leggio, contributo di "Ingegneria del Software" al Manuale di Informatica, Calderoni, 1993
D. Natale, Qualità e Quantità nei Sistemi Software, FrancoAngeli, 1995
D. Natale, Un nuovo metro per il software, Zerouno n. 161, 1995
IFPUG-GUFPI, Software Measurement and Management, International Conference 1996, Roma.
C. Jones, Software System Failure and Success, Thomson Computer Press, 1996
C. Jones, Applied Software Measurement, McGraw Hill, 1997
F. La Noce, Function Point, FrancoAngeli, 1997
L. Buglione, Misurare il Software, FrancoAngeli, 1999-2003
IFPUG, Function Point: Manuale delle Regole di Conteggio – Versione 4.1.1, 2000
D. Garmus – D. Herron, Function Point Analysis: Measurement Practices for Successful Software Projects, Addison-Wesley, 2000
B. W. Boehm e altri autori, Software Cost Estimation With Cocomo II, Prentice-Hall, 2000
K. Thompson, Advanced Function Point Analysis Workshop, Technology Transfer, Roma 3-5 dicembre 2001
M. Hotle, Function Point can help Measure Application Size, Gartner, 2002
IFPUG, Bigger and Better Metrics, Conference 2002, USA
D. Lipton, The Psychology of Function Point Counting, Q/P Management Group Inc., 2003
S. Goldfarb, Establishing a Measurement program, Q/P Management Group Inc., 2003
IFPUG, Don't Sweat the Metrics, Conference 2003, USA

Webgrafia

- Gruppo Utenti Function Point Italia
www.gufpi.org
Ente di normazione UNINFO, National Body ISO, federato all'UNI contenente norme e riferimenti
www.uninfo.polito.it
International Function Point User Group
www.ifpug.org
International Software Benchmarking Standard Group
www.isbsg.org.au

The Common Software Measurement International Consortium

www.cosmicon.com

United Kingdom Software Metrics Association

www.uksma.co.uk

Netherlands Software Metrics Association

www.nesma.nl

Deutschsprachige Anwendergruppe für Software-Metrik und Aufwandschätzung

www.dasma.de

Asociación Española de Métricas del Software

www.aemes.fi.upm.es

Brazilian Function Point Users Group

www.bfpug.br

Korea Function Point Users Group

www.kfpug.co.kr

Australian Software Metrics Association

www.asma.org.au

Personal website - CFPS expert

www.functionpoints.com

Pagine sui Function Point in italiano dal sito personale dell'autore

www.mondomatica.it/function.htm

4. L'associazione GUFPI-ISMA, i Function Point e le metriche funzionali

Intervento di Loredana Mancini - Presidente GUFPI-ISMA

(con la collaborazione di Luigi Buglione, Coordinatore SMC – Software Measurement Committee, e Roberto Meli, Coordinatore CPC – Counting Practice Committee)

1. L'associazione GUFPI-ISMA e le sue finalità

L'associazione denominata "GUFPI - Italian Software Metrics Association", di seguito "GUFPI-ISMA", è nata come associazione senza scopo di lucro con l'obiettivo di mettere in contatto esperti nel settore metriche, l'idea di avvio del gruppo di lavoro è stata quella di organizzare un punto di incontro per quegli esperti e quelle aziende che stavano avviando al proprio interno la sperimentazione dell'uso dei Function Point e in particolare stavano valutando l'adozione di metriche per la gestione e pianificazione di progetti software. I soci fondatori Getronics e SOGEI, utilizzavano già al proprio interno tale metrica e quindi rappresentavano un corretto punto di aggregazione sull'argomento.

Ad oggi il GUFPI-ISMA, nata nel corso del 1989, vede l'adesione di circa 50 società come soci partecipanti e sostenitori

L'Associazione, costituita senza fini di lucro, si propone come scopi:

- la promozione e lo sviluppo nel territorio italiano delle tecniche quantitative di misurazione dei prodotti, processi e servizi ICT ed in particolare delle tecniche di misura funzionale del software come i Function Point, attraverso la collaborazione, la ricerca e lo scambio di esperienze tra i soci partecipanti;

- la promozione, lo sviluppo e il coordinamento di attività volte all'aggiornamento culturale e professionale tra i suoi soci;
- la preparazione, l'organizzazione e la partecipazione a meetings, seminari di studio e manifestazioni, anche all'estero, con l'utilizzo di attrezzature e/o materiali propri o altrui;
- la promozione e lo sviluppo delle attività dell'Associazione attraverso la pubblicazione e/o distribuzione, anche in via digitale e telematica, di periodici specializzati, di manuali, nonché del materiale e della documentazione tecnica prodotta a beneficio dei soci e di tutti gli interessati;
- lo studio e la valorizzazione delle attività dell'Associazione attraverso l'adesione ad altre associazioni internazionali aventi analoghi scopi, quali IFPUG o ISBSG nonché attraverso la collaborazione con organismi internazionali come UE, IEEE, ISO, UNI, UNINFO;
- stipulare convenzioni con enti pubblici o privati per la gestione di corsi e seminari, nonché garantire la prestazione di servizi rientranti nell'ambito dei propri scopi istituzionali.

Le attività del gruppo si svolgono con forme di attività volontarie da parte dei soci e sono previsti sia gruppi di lavoro temporanei finalizzati allo studio di argomenti specifici sia gruppi di lavoro stabili per l'analisi continua e l'aggiornamento delle tecniche utilizzate.

2. Comitati Tecnici

Per svolgere proficuamente le proprie attività tecniche il GUFPI-ISMA ha alla data 3 gruppi di lavoro attivi:

2.1. CPC – Counting Practice Committee

Il Counting Practices Committee (CPC) è un organismo interno al GUFPI-ISMA ed agisce su sua delega negli ambiti che gli sono propri, il coordinatore di questo gruppo è Roberto Meili. Gli obiettivi generali del CPC sono i seguenti:

- Contribuire al miglioramento della formulazione delle regole di conteggio relative alla metrica dei Function Point.
- Favorire una omogenea interpretazione delle regole stesse a livello interaziendale.
- Contribuire al processo di diffusione e promozione degli standard a livello nazionale.
- Favorire l'interscambio informativo con le organizzazioni internazionali che hanno obiettivi analoghi.

Per raggiungere i propri obiettivi il CPC può:

- Verificare la correttezza e la completezza delle regole emanate dall'IFPUG in italiano, in inglese e dei case studies relativi.
- Analizzare le problematiche suggerite dagli utenti di tale metrica.
- Raccogliere e diffondere le indicazioni provenienti dal CPC internazionale IFPUG (Internationale Function Point Users Group).
- Proporre correzioni, miglioramenti ed estensioni delle regole all'IFPUG, sia in versione inglese, sia nella traduzione italiana.
- Definire e diffondere le Linee Guida Italiane di supporto alla lettura delle regole.
- Stabilire gli opportuni contatti con altri organismi.

2.2. SBC – Software Benchmarking Committee

Il Comitato sul Software Benchmarking riunisce i membri del GUFPI-ISMA che sono interessati all'approfondimento delle tecniche di standardizzazione usate per confrontare diverse performance con particolare riferimento alle produttività e costi unitari. Il coordinatore di questo gruppo è Domenico Natale e Luca Santillo.

Il SBC si propone di:

- aggiornare i membri sulla letteratura di settore e raccogliere riferimenti metrici pubblici;
- promuovere la raccolta dei dati;
- mantenere i contatti internazionali con l'ISBSG (International Software Benchmarking Standards Group) collegato con il comitato sul Benchmarking dell'IFPUG (International Function Point Users Group);
- stabilire accordi con simili attività di ricerca svolte in ambito accademico.

È in via di definizione uno studio di ausilio per la lettura dei vari coefficienti di produttività e costi unitari disponibili i quali non possono essere utilizzati correttamente, a certi livelli di dettaglio, se non in co-presenza di specifici attributi di riferimento, tra cui:

- dimensione dell'applicazione software
- metodo di calcolo della dimensione
- metodologia
- case tool
- linguaggio
- piattaforma tecnologica
- processo/fase
- criticità
- complessità
- qualità attesa
- contesto economico/organizzativo
- nazione

2.3. SMC - Software Measurement Committee

Nel Febbraio 2001 è stato costituito un nuovo Gruppo di Lavoro, denominato SMC (Software Measurement Committee) [SMC03a], con l'obiettivo di occuparsi degli aspetti di misurazione da un punto di vista non funzionale, trattando aspetti organizzativi quali piani di misurazione, iniziative di miglioramento del processo di sviluppo del software (software process improvement), tool metrici e misurazione delle performance del software, in ossequio al dettato dello statuto GUFPI-ISMA . I coordinatori di questo gruppo sono Luigi Buglione e Claudio Grande.

In Figura 1 si evidenziano i quattro possibili ambiti di azione ciascuno dei domini di interesse del SMC (il primo dei quali è quello degli ambienti web-based):

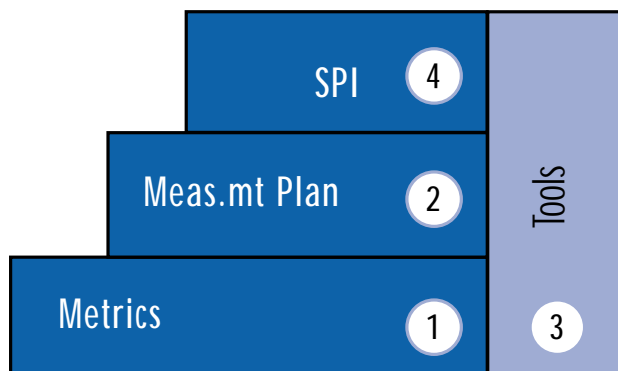
1. analisi delle metriche ed indicatori
2. elaborazione di un piano di misurazione usando la regola del "5W's+H" (What, why, who, where, when, how) [PFLE93]
3. dei tool automatici che permettano di poter minimizzare l'effort per raccolta ed analisi dei dati

4. elaborazione-analisi di modelli di miglioramento di processo

Per quanto concerne la misurazione legata ad ambienti basati sul web, il punto di partenza è dato dal cercare di utilizzare un approccio non funzionale per la misurazione del web. Adattamenti della FPA [LONG] o nuovi metodi che modellano la FPA alle entità tipiche del Web (es: multimedia, streaming, ...), come WebObjects [REIF00] non esauriscono però la totalità delle prospettive possibili. Difatti il web non necessariamente deve essere visto dal punto di vista del programmatore; un'altra prospettiva è quella di valutare per un ambiente web la sua usabilità dal punto di vista di un "first-time" visitor o la sicurezza offerta da una transazione e-business dal punto di vista di un potenziale acquirente on-line [WEBQ00], tutti aspetti non facilmente catturabili adattando la logica funzionale ad oggetti con una natura diversa.

Il primo deliverable del SMC è stata pertanto la definizione di un Web Quality Model (WQM) [SMC03b] basato su tre livelli (caratteristiche,

Fig. 1 – Ambiti di azione del SMC



sotto-caratteristiche, metriche), in modo analogo allo standard ISO/IEC 9126-1:2001 [ISO01], non applicabile però "as-is" a catturare la diversità e complessità di punti di vista tipici negli ambienti web-based.

3. I Function Point

Il problema della stima e quantificazione delle dimensioni di un sistema software è sempre stato l'anello debole dell'attività di pianificazione impegni e gestione progetti. Le metriche spesso utilizzate si basavano su una stima delle dimensioni del software utilizzando il concetto di Linea di Codice (LOC Line of Code), nelle sue diverse interpretazioni, LOC, SourceLOC, ecc.

Come si può facilmente intuire tale tipo di misurazione, ha in sé il problema della standardizzazione, infatti il conteggio delle linee di codice può dipendere sia dal linguaggio, sia dall'ambiente di sviluppo, sia dalle modalità di conteggio.

Inoltre negli attuali sistemi di sviluppo le attività di integrazione rappresentano un fattore sempre più preponderante rispetto alle pure attività di sviluppo o manutenzione del software. Proprio per superare tali limiti è stata studiata da Allan J. Albrecht una metodologia denominata Function Point Analysis (FPA). I F.P. sono una metrica di tipo funzionale che cerca, tramite oggetti propri, di individuare le dimensioni di un sistema quantificando le funzioni consegnate all'utente e quindi forniscono una quantificazione di livello superiore rispetto alla rappresentazione tecnica. Dagli iniziali F.P. definiti da Albrecht la metrica funzionale ha subito molte evoluzioni e oggi esistono organismi di standardizzazione a livel-

lo internazionale che hanno il compito di far evolvere questa tipologia di approcci e di superarne le eventuali criticità di applicazione e d'uso.

4. Le evoluzioni delle metriche funzionali (a cura di Luigi Buglione)

Le applicazioni della FPA nel corso dei primi anni '80 hanno dato luogo ai primi commenti sul suo corretto e pieno ambito di applicazione, identificando alcune criticità [SYMO88] [DESH89] [KITC97]:

- **metodologia pensata per il mondo MIS:** non tiene conto del mondo Real Time e TLC in genere;
- **eccessiva semplificazione delle tipologie dei componenti:** ad un sistema che contiene più di 100 DET viene attribuito, al massimo, il doppio dei punti di un componente con un solo DET;
- **bassa granularità degli intervalli di complessità:** ad interventi minimi di manutenzione adattiva del software corrisponde uno stesso numero di FP sia per un elemento che per 19 elementi modificati (nel caso degli EIF);
- **scelta dei pesi per i componenti:** è criticabile la mancata revisione negli anni dei singoli pesi, considerando che le ponderazioni originarie sono altresì frutto dell'esame di 22 progetti assolutamente non rappresentativi di diversi;
- **intransitività dell'addizione nel calcolo delle FP:** il punteggio complessivo di n piccoli progetti è maggiore di quello della singola applicazione comprendente gli n progetti, a causa dell'inclusione in più di (n-1) volte dei relativi ILF e/o EIF riconteggiati. L'IFPUG ha comunque preso nota di tale problema; campi di azione;

- **grado di influenza delle GSC:** l'attribuzione di un valore soggettivo - e quindi qualitativo - contrasta con la propugnata oggettività. Ancora, la restrizione a sole 14 GSC probabilmente non soddisfa la generalità dei casi, rendendosi necessario quindi un aggiornamento del numero e della natura delle stesse, secondo le evoluzioni del mondo informatico;
- **insufficiente approfondimento dei dettagli** sul contenuto delle mappe, dei report dei dati. Ciò può generare errori di valutazione qualora si tendesse a riutilizzare in modo spinto moduli di vario genere;
- **automazione** – last but not least, sicuramente uno dei motivi che ha frenato la diffusione e l'utilizzo della FPA è stato quello dell'automazione, da riferirsi a due possibili direzioni: quella dell'archiviazione e analisi dei dati derivati dal conteggio e quella relativa al conteggio automatico stesso.

Nel corso degli anni una serie di varianti dello studio originario di Albrecht sono state create, dando luogo alla branca della misurazione del software che attualmente viene definita **Functional Size Measurement (FSM)**. I tre principali metodi FSM sono, in ordine di introduzione temporale:

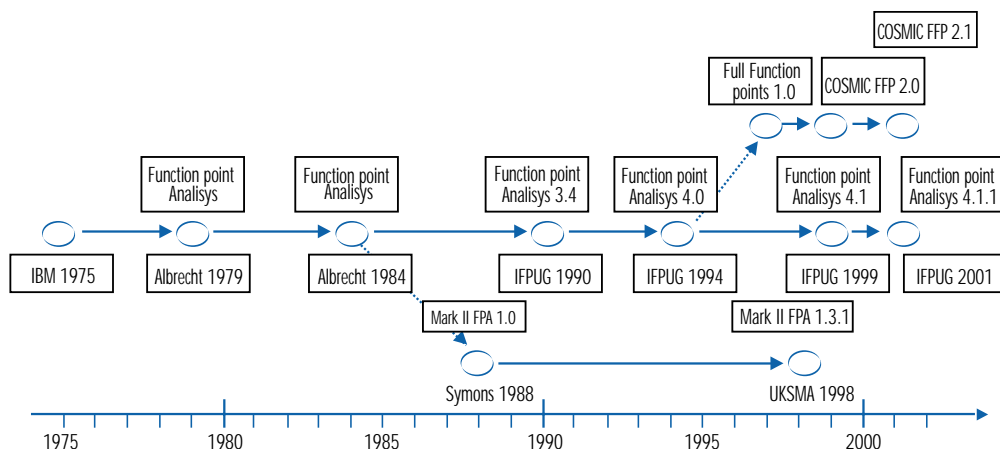
- IFPUG Function Point Analysis (**FPA**) [IFPU99]
- UKSMA **Mark II FPA** [USKMA98]
- COSMIC Full Function Points (**FFP**) [ISO03]

La figura seguente traccia un quadro di insieme dell'evoluzione temporale relativa ai principali metodi FSM.

5. Lo standard ISO 14143

La rilevanza assunta negli anni dalla FPA, assunto a standard de facto nel mondo dell'Ingegneria del Software, ha condotto negli ultimi anni l'ISO a considerare le metriche funzionali quali prossimi standard de jure. Un nuovo Working Group (12) del JTC1/SC7 è stato difatti istituito nel 1997 con l'obiettivo di "Stabilire un set di standard pratici per la misurazione funzionale del software (FSM). FSM è un termine generico per indicare metodi di dimensionamento del software basati su una visione esterna del prodotto, ed include metodi come la Function Point Analysis". Lo standard previsto per dare risposta a tale obiettivo è il 14143, uno standard multi-parte (finora sono 5 le parti previste):

Fig. 2 – Evoluzione dei metodo di FSM (BUGL03)



- **14143-1:1998** (Definition of Concepts). Unica parte già promulgata come standard internazionale (IS), ridefinisce in forma generica i concetti introdotti da Albrecht;
- **14143-2:2003** (Conformity evaluation of software size measurement methods to ISO/IEC 14143-1:1998). Si propone una valutazione di conformità dei vari Software Sizing Methods attraverso una checklist le cui domande debbono rispondere a tutti i requisiti fissati dalla 14143-1 e produrre, in caso positivo, un documento che ne ufficializzi tale conformità;
- **14143-3:2003** (Verification of Functional Size Measurement Methods). Intende assicurare, partendo dal modello di Jacquet e Abran [JACQ97] [JACQ99], che il metodo sotto esame permetta:
 - Ripetibilità e riproducibilità;
 - Accuratezza;
 - Convertibilità;
 - Soglie di discriminazione;
 - Applicazione ai vari domini funzionali.
- **14143-4:2002** (Functional Size Measurement Reference Model). Si illustra il modello di riferimento per poter effettuare una quantificazione della dimensione di un software appropriata, visto la semplice dichiarazione di conformità espressa sulla base dei requisiti della 14143-1 potrebbe non essere sufficiente. A tal fine lo standard fissa una serie di RUR (Reference User Requirements), usabili in diversi domini funzionali e metodi FSM, per minimizzare la difformità nelle valutazioni.
- **14143-5** (Determination of Functional Domains for Use with Functional Size Measurement). Tale parte riguarda la determinazione dei domini funzionali da utilizzare con un FSM, proponendosi quale complemento al ISO/IEC TR 12182:1998 intitolato “Cate-

gorization of Software”. Per dominio funzionale (o functional domain – FD) si intende una classe di software basata sulle caratteristiche di un requisito funzionale espresso dall’utente (FUR – functional user requirement), pertinenti al FSM. La Sezione 5 fissa i requisiti generali per poter identificare un FD, mentre gli Allegati A, B e C (informativi) propongono tre distinti modelli di identificazione con approcci – nell’ordine – di tipo “top-down” (basato su un’estesa analisi della letteratura tecnica esistente, utile per associare i FUR ai relativi FD), “bottom-up” (basato sul concetto di layer, come nei COSMIC--FFP, per definire un set di BFC e costruire un modello FSM utile a definire i vari FD) e “quantitativo” (basato su una tripartizione di FD secondo la “ricchezza” di dati, movimenti e algoritmi).

Recentemente si è dato luogo ad una sesta parte (14143-6) relativa ad una “Guide for use of ISO/IEC 14143 series and related international Standards”.

6. COSMIC-Full Function Points

Una delle critiche maggiori alla FPA nella formulazione tipica gestita attualmente dall’IFPUG è quella relativa alla copertura di ambienti prettamente MIS, non catturando l’essenza di ambienti Real/Time e TLC. Da questa esigenza sono nati nel 1997 i Full Function Points (FFP) [STPI97] [DESH97], affiancano ai tradizionali “Management Processes” e alle 5 ben note componenti dati (ILF, EIF) e transazionali (EI, EO, EQ) la categoria dei processi di controllo (Control Processes), caratterizzata da 4 nuovi controlli di tipo transazionale (ECE – External Control Entry, ECX – External Control eXit, ICR – Internal Control Read e ICW

- Internal Control Write) e 2 nuovi controlli di tipo dato (UCG - Updated Control Group e RCG - Read-Only Control Group), mantenendosi però in linea con la struttura generale dell'IFPUG. La seguente tabella raffronta gli elementi strutturali nei due metodi:

Concetti	Full Function Points	IFPUG FPA
Boundary	Boundary	Boundary
Users	Users	Users
Data Objects	Groups of Control Data	Logical Files
Process Objects	Control Process	Elementary Process
Sub-process Objects	Sub-Process	N/A

Negli FFP si evidenziava quindi un maggior livello di dettaglio rispetto agli IFPUG FPs, dovuto alla definizione e valutazione dei sottoprocessi, non presenti nella versione IFPUG e quindi una maggiore granularità nel conteggio. La procedura e i passi da eseguire rappresentano una personalizzazione di quanto proposto dall'IFPUG, con l'aggiunta delle fa-

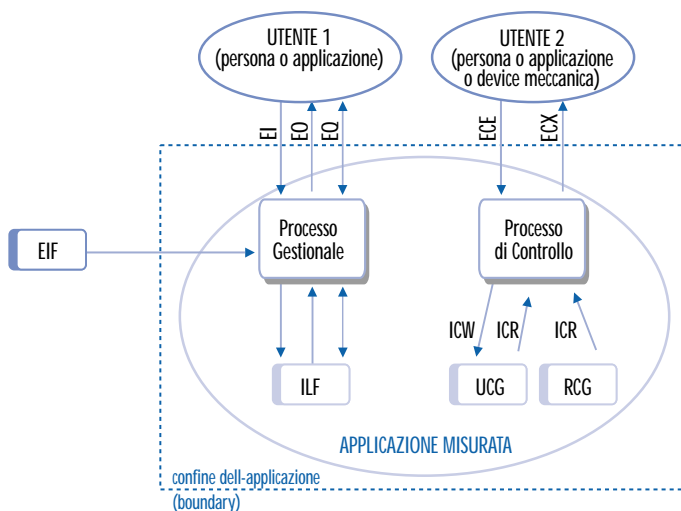
si di calcolo relative ai processi di controllo. L'evoluzione dei FFP, nata nel 1998 e gestita da un consorzio internazionale denominato COSMIC (Common Software Metric International Consortium), è stata denominata per l'appunto COSMIC-FFP.

Il COSMIC-FFP method rappresenta quindi non più una evoluzione dei FFP, bensì un nuovo metodo FSM indipendente, estendendo quei concetti introdotti con FFP a tutti i domini applicativi (sebbene la numerazione delle versioni inizi con la v2.0, a dare una continuità, visti gli autori principali, ai concetti espressi con il FFP method). Viene spesso referenziato come una "seconda generazione di metodi di misurazione funzionali".

Il metodo, nel pubblico dominio, è divenuto recentemente standard ISO (ISO/IEC 19761:2003), assumendo pertanto - non solo dal punto di vista tecnico - una valenza di sicuro interesse per la comunità del Software Engineering.

Nella versione 2.0 [ABRA99], le differenze principali tra COSMIC-FFP e FFP sono rinvenibili in:

Fig. 3 - Tipi Funzione nella FFP v1.0



- Applicabilità del metodo a tutti i domini applicativi (MIS, R/T, Ibridi);
- componenti dati non contribuiscono più al valore finale di FFP;
- Fissazione de peso dei sotto-processi ad 1, secondo il numero di DET dei singoli elementi;

Ridenominazione delle componenti transazionali in Functional Process Types, rappresentati da quattro possibili data movements: Entry, Exit, Read e Write, che generalizzano i quattro tipi di sotto-processo presenti in FFP v1.0 (ECE, ECX, ICR, ICW).

I concetti fondanti in COSMIC-FFP sono quattro:

- Layer: diversamente dagli altri metodi FSM, COSMIC-FFP considera il software come una serie di strati (layers). Ogni layer incapsula funzionalità utili per il layer che usa i suoi servizi e usa le funzionalità fornitegli dal layer di livello inferiore, permettendo di conteggiare il numero di Cfsu (COSMIC functional sizing unit) al livello di granularità desiderato l'ambiente in esame.
- Boundary: esiste un confine implicito tra ogni layer identificato sulla base dei FUR (Functional User Requirements).

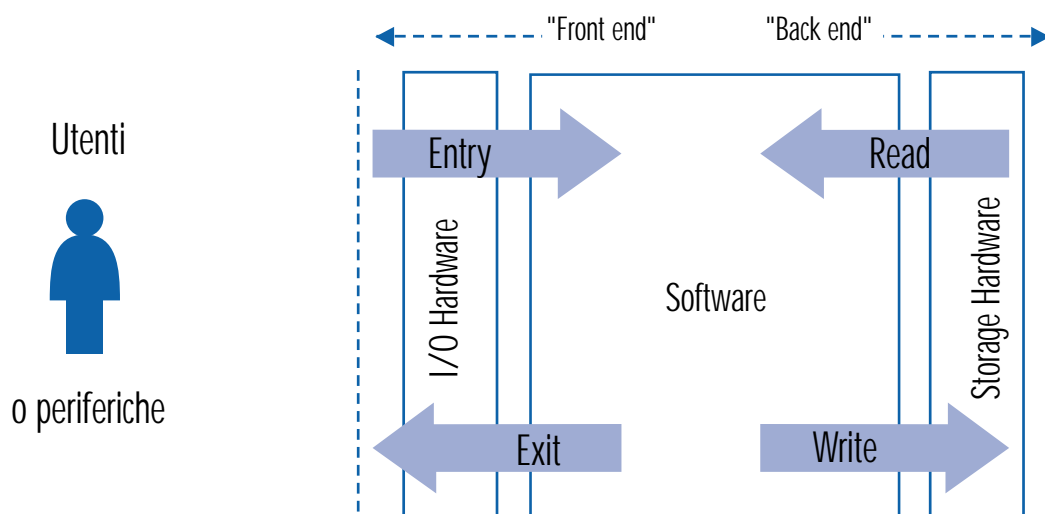
- Software Users: si estende il concetto di user non solo a persone, ma anche ad apparecchiature (come già specificato nella FPA, MkII e FFP) o altro software.

Si stabilisce ancora che porzioni di software dentro i layer immediatamente circostanti sono considerate utenti quando interagiscono con la porzione misurata.

- FUR – Functional User Requirements: espressione creata in ambito ISO per descrivere un sotto-insieme di requisiti utente, catturati dal punto di vista dell'utente. I FUR rappresentano quindi semplicemente le azioni e le procedure che il software dovrà eseguire per rispondere ai bisogni dell'utente, escludendo i requisiti di tipo qualitativo o tecnico.

La versione 2.1 [ABRA01], rilasciata nel 2001, ha rappresentato invece una versione minore, con l'obiettivo di allineare il metodo quanto più possibile ai precetti della famiglia di standard ISO/IEC 14143, chiarendo inoltre – laddove possibile – le regole di conteggio. La recente versione 2.2 [ISO03], rilasciata nel Gennaio 2003 con il sottotitolo di "The CO-

Fig. 4 – Flusso dei dati secondo COSMIC-FFP



SMIC Implementation Guide For ISO/IEC 19761: 2003”, chiarifica ulteriormente le regole di conteggio, a seguito dei feedback provenienti dai Trials worldwide ed allinea la terminologia in modo definitivo con la versione ISO.

Un punto di sicuro interesse è dato dalla comparabilità e conversione tra conteggi effettuati con diversi metodi FSM. Uno studio di riferimento è sicuramente il technical report di Thomas Fetcke [FETC99] dell'Università Berlino . In particolare, i metodi comparati sono i tre principali FSM: IFPUG FPA (versioni 4.0 e 4.1), MkII FPA (versione 1.3.1), FFP (versione 1.0) e COSMIC-FFP (versione 2.0). Il case study evidenzia, attraverso attraverso la misurazione di un fittizio warehouse software portfolio composto da 5 applicazioni (denominate W, M, C, LC, LS), quanto i metodi siano fortemente corrispondenti, tanto dal punto di vista architetturale, quanto dimensionale. La seguente tabella riassume i risultati dei conteggi effettuati con i diversi metodi FSM.

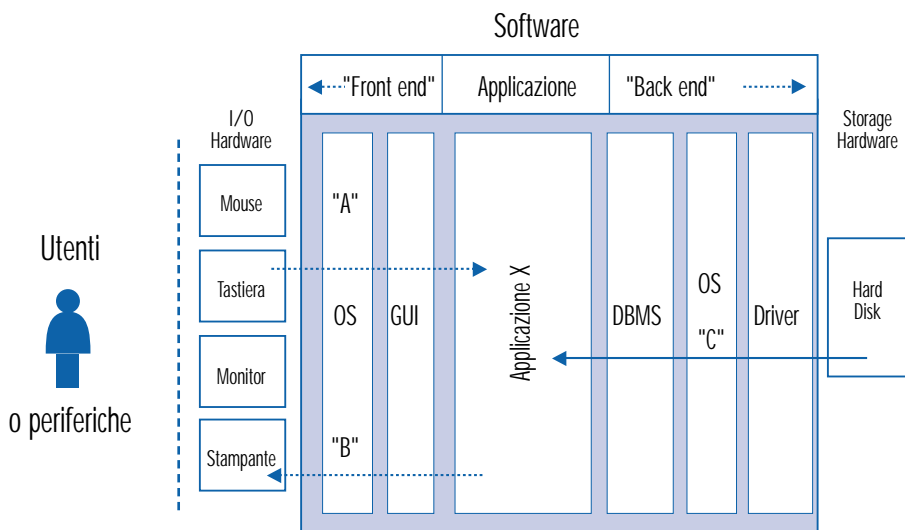
Tab. 2
Risultati finali conteggio comparato [FETC99]

	Metodo	IFPUG		MkII	FFP	COSMIC
Appl.	# Trans.	4.0	4.1	1.3.1	1.0	2.0
W	16	78	77	72.96	102	81
M	8	40	40	32.40	52	38
C	10	50	49	46.72	65	51
LC	10	57	56	48.96	71	52
LS	6	31	31	24.00	41	29

Una questione delicata, sottolineata altresì dall'oggetto del Capitolo 6 del Manuale di Misurazione di COSMIC-FFP v2.2 (COSMIC-FFP Convertibility), è quella relativa alla eventuale esistenza di fattori di conversione tra i diversi metodi FSM, al pari di quanto effettuato con il backfiring tra LOC e FPs.

La guida (che considera in particolare IFPUG CPM 4.1, Mark II CPM 1.3.1 e FFP v1.0) fornisce una procedura composta da quattro passi, sottolineando la necessità di:

Fig. 5 – Flusso dei dati per un software MIS secondo COSMIC-FFP



- sviluppare formule di conversione “locali” (basate sul proprio database storico dei dati di progetto);
 - conteggiare dallo stesso punto di vista con i diversi metodi;
 - eliminare eventuali outliers che potrebbero
- inficiare la bontà del R2, cercando di determinare le motivazioni per tali scostamenti. Ovviamente l'eventuale backfiring dovrebbe rappresentare solo un momento di passaggio e non un passo sistematico per evitare conteggi nativi con un dato metodo (e/o versione) di conteggio.

7. Bibliografia

- [ABRA99] ABRAN A., DESHARNAIS J.M., OLIGNY S., ST-PIERRE S. & SYMONS C., COSMIC-Full Function Point Measurement Manual, Common Software Measurement International Consortium, Version 2.0 (Field Trial Version), October 1999
- [ABRA01] ABRAN A., DESHARNAIS J.M., OLIGNY S., ST-PIERRE S. & SYMONS C., COSMIC-Full Function Point Measurement Manual, Common Software Measurement International Consortium, Version 2.1, May 2001, URL: <http://www.lrgl.uqam.ca/cosmic-ffp>
- [BUGL03] BUGLIONE L., Misurare il software. Dimensione, qualità, standard e miglioramento di processo nell'ICT, FrancoAngeli, Collana Informatica & Organizzazioni, 742.20, 2/e, 2003, ISBN: 88-464-4634-8, URL: www.geocities.com/lbu_measure/libri/mis.htm
- [DESH89] DESHARNAIS J.M., FPA: counting rules with the “entity” concept, IFPUG 1989 Spring Conference Proceedings, April 3-6, 1989, San Diego, CA
- [DESH97] DESHARNAIS J.M., ST-PIERRE D., MAYA M. & ABRAN A., Full Function Points: Counting Practices Manual - Procedures and Counting Rules, SERML/SELAM, November 1997, URL: <http://www.lrgl.uqam.ca/publications/pdf/257.pdf>
- [FETC99] FETCKE T., The Warehouse Software Portfolio. A Case Study in Functional Size Measurement, University of Berlin, Technical Report, 1999-20, 1999 – URL: <http://www.lrgl.uqam.ca/cosmic-ffp/casestudies/Fetcke1999b.pdf>
- [IFPU99] IFPUG, Function Point Counting Practices Manual, v.4.1.1, Chairperson: Mary Bradley, January 1999; URL: <http://www.ifpug.org/publications/manual.htm>
- [ISO01] ISO/IEC JTC1/SC7, IS 9126-1 Information Technology – Software quality characteristics and metrics Part 1 – Quality characteristics and sub-characteristics, 2001; URL: <http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=22749&ICS1=35&ICS2=80&ICS3=>
- [ISO03] ISO/IEC JTC1/SC7, IS 19761:2003 - Software Engineering - Functional Size Measurement – COSMIC-FFP Functional Size Measurement Method, January 2003; URL: <http://www.lrgl.uqam.ca/cosmic-ffp/manual.html>
- [JACQ97] JACQUET J.P. & ABRAN A., From Software Metrics to Software Measurement: A Process Model, The 3th IEEE International Software Engineering Standard Symposium, ISESS'97, Walnut Creek, CA, June 2-6 1997, IEEE Computer Society Press, URL: <http://www.lrgl.uqam.ca/publications/pdf/208.pdf>

- [JACQ99] JACQUET J.P. & ABRAN A., Metrics Validation Proposals: A Structured Analysis, Dumke, R., and Abran, A. (eds.): Software Measurement, Gabler, Wiesbaden, 1999, 43-60, URL: <http://www.lrgl.uqam.ca/publications/pdf/361.pdf>
- [KITC97] KITCHENHAM B., The Problem with Function Points, IEEE Software, IEEE Computer Society, Vol. 14 No. 2, March/April 1997, pp. 29-30, URL: <http://csdl2.computer.org/dl/mags/so/1997/02/s2029.pdf>
- [LONG] LONGSTREET D., Function Points applied to New Technologies, URL: <http://www.softwaremetrics.com/files/newtechenglish.zip>
- [OLIG99] OLIGNY S. & ABRAN A., On the Compatibility between Full Function Points and IF-PUG Function Points, Proceedings of the 10th European Software Control and Metric Conference (ESCOM-SCOPE 99), Herstmoreux Castle, UK, October 1999, URL: <http://www.lrgl.uqam.ca/publications/pdf/431.pdf>
- [PFLE93] PFLEEGER S.L., Lessons learned in building a corporate metrics program, IEEE Software, IEEE Computer Society, Vol. 10 No. 5, May 1993, pp. 67-74; URL: <http://csdl2.computer.org/dl/mags/so/1993/03/s3067.pdf>
- [REIF00] REIFER D., Web Development: Estimating Quick-to-Market Software, IEEE Software, November/December 2000, pp. 59-64; URL: <http://csdl.computer.org/comp/mags/so/2000/06/s6057abs.htm>
- [SMC03a] SMC, Documento Istitutivo CT (001.02), Internal Document, GUFPI Software Measurement Committee, May 2003; URL: <http://www.gufpi.org/smc.htm>
- [SMC03b] SMC, Guida Misure & Metriche (001.04), Internal Document, GUFPI CT Software Measurement Committee, May 2003; URL: <http://www.gufpi.org/smc.htm>
- [STPI97] ST.PIERRE D., MAYA M., ABRAN A., DESHARNAIS J.M. & BOURQUE P., Full Function Points: Counting Practices Manual, SERML/SELAM, Technical Report 1997-04, September 1997
- [SYMO88] SYMONS C.R., Function Point Analysis: Difficulties and Improvements, IEEE Transactions on Software Engineering, IEEE Computer Society, Vol. 14, No. 1., January 1988, pp. 2-10; URL: <http://csdl.computer.org/comp/trans/ts/1988/01/e0002abs.htm>
- [UKSMA98] UKSMA, Mk II Function Point Analysis Counting Practices Manual, Version 1.3.1, United Kingdom Software Metrics Association Metrics Practices Committee, Edenbridge (UK), September 1998, URL: <http://www.uksma.co.uk/public/mkIIr131.pdf>
- [WEBQ00] BARNES, S. J. AND R. T. VIDGEN, "WebQual: An Exploration of Web Site Quality," Proceedings of the Eighth European Conference on Information Systems, Vienna, July 3-5, 2000, URL: <http://www.webqual.co.uk/papers/318.pdf>

5. Analisi di tecniche di stima dei costi di sviluppo del software

Sandro Morasca - Università degli Studi dell'Insubria, Dipartimento di Scienze Chimiche, Fisiche e Matematiche (sandro.morasca@uninsubria.it)

Abstract

Una stima accurata dei costi di sviluppo del software può dare alle organizzazioni software un vantaggio competitivo. Tuttavia, a causa dei numerosi fattori che possono influenzare lo sviluppo del software, non si è ancora giunti a un modello consolidato per la stima dei costi del software. Diverse tecniche sono state proposte in letteratura e utilizzate industrialmente. L'obiettivo della presentazione è di analizzare criticamente tali tecniche e di mostrarne punti di forza e debolezza in modo tale da fornire elementi per prendere decisioni su quali tecniche sono più facilmente e utilmente applicabili in specifiche applicazioni.

1. Introduzione

Un'accurata stima dei costi per i progetti software può portare svariati benefici sia a un'organizzazione che costruisce software sia a un'organizzazione che acquisisce software. In particolare, la stima dei costi di un progetto software è utile (queste sono alcune tra le motivazioni più importanti) per

- stipulare contratti di sviluppo sensati (sia dal punto di vista del costruttore sia dal punto di vista dell'acquirente);
- prevedere tempi di sviluppo ragionevoli (sia dal punto di vista del costruttore sia dal punto di vista dell'acquirente):
 - il time-to-market è importante sia per sviluppi su commessa sia per sviluppi interni o di prodotti;
- dimensionare il team di sviluppo:
 - partire con un team sottodimensionato genera ritardi e prodotti di qualità scadente;
 - gli sviluppatori tenderanno comunque a consumare tutto il tempo a disposizione.

Diversi sono i fattori che concorrono a formare il costo del software. Tuttavia, in un'attività a così forte intensità di partecipazione umana quale lo sviluppo del software, il costo è legato principalmente al costo delle risorse umane, più che al costo di attrezzature informatiche o a costi generali di struttura. Pertanto, nel seguito l'enfasi sarà incentrata sullo sforzo, ovvero la quantità di lavoro che viene impiegata nello svolgimento di un progetto software.

Molti e di natura diversa sono i fattori che influenzano lo sforzo necessario per eseguire un progetto software. Alcuni, comuni a molti ambienti, sono qui riportati.

- **Dimensione del software:** tale fattore sembra essere comunemente accettato come quello principale e viene utilizzato da molti modelli di stima dei costi.
- **Fattori umani:** la produttività dipende fortemente dalle capacità delle risorse umane impiegate.

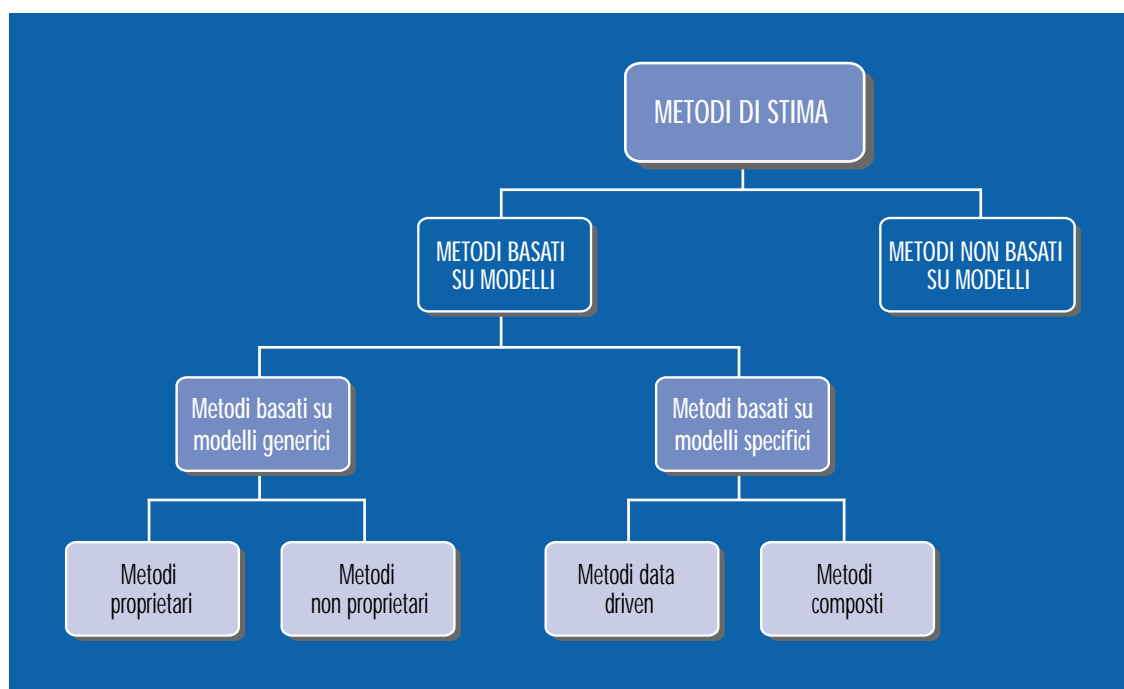
- **Complessità dell'applicazione:** a parità di tutti gli altri fattori, applicazioni diverse possono avere livelli diversi di difficoltà.
- **Stabilità dei requisiti:** è senz'altro uno dei fattori più critici, soprattutto per i problemi che possono essere causati da variazioni dei requisiti a progetto già avviato, specialmente quando il progetto sia già in fase avanzata.
- **Prestazioni e vincoli non funzionali richiesti all'applicazione:** vincoli stringenti e prestazioni elevate implicano costi più elevati.
- **Ambiente di sviluppo:** il costo può variare notevolmente in dipendenza delle caratteristiche dell'ambiente, dell'organizzazione del team di sviluppo etc.

La stima dei costi del software è un'attività che è stata effettuata da quando il software ha assunto un'importanza rilevante per gli altri settori industriali ed è perciò diventato esso stesso un settore trainante dell'economia. Ciò ha portato a una proliferazione di modelli, che sono stati introdotti per esigenze e ambienti applicativi diversi. L'obiettivo del presente documento è di illustrarli nelle loro linee essenziali, identificarne similarità e differenze e discuterne pregi e difetti in maniera comparativa, per fornire elementi a chi debba scegliere quale modello o quali modelli adottare.

2. Schema di classificazione

Dato il numero elevato di metodi per la stima dei costi, può essere utile effettuare una classificazione [BW00] al fine di identificarne similarità e differenze, mostrata in figura 1.

Figura 1. Classificazione dei metodi.



Stima basata su modelli generici

In genere questi modelli sono ritenuti applicabili a contesti diversi.

- Metodi proprietari. I dettagli relativi alle modalità di stima non sono pubblici.
- Metodi non proprietari. La meccanica interna del modello di stima dei costi è pubblica

Metodi basati su modelli specifici

Sono basati su modelli locali la cui validità si estende al contesto in cui sono stati sviluppati.

- Metodi data driven. I modelli vengono derivati a partire dai dati, con tecniche statistiche oppure di machine learning. I modelli possono essere detti "parametrici" se sono costruiti in base a una preconstituita forma funzionale, ad esempio lineare, oppure "non parametrici" se invece non si suppone la sussistenza di una precisa forma funzionale, come quando si suppone che vi sia una relazione monotonamente crescente tra la dimensione del software e il suo costo.
- Metodi composti. I modelli vengono costruiti in base all'opinione di esperti e tecniche di modellazione data driven. Il metodo di modellazione descrive come applicarle e combinarle per costruire un modello di stima.

Metodi non basati su modelli

Consistono di una o più tecniche di stima e di una specifica di come applicarli in uno specifico contesto, e non implicano la costruzione di modelli, ma direttamente di una stima. Tali metodi tipicamente richiedono la consultazione di uno o più esperti per derivare una stima dello sforzo che è di tipo soggettivo. Lo sforzo può essere determinato in maniera bottom-up oppure top-down, in base a una scomposizione per componenti oppure per attività. Inoltre, può essere effettuata anche una stima dello sforzo d'integrazione dei componenti oppure delle attività., in modo da ottenere una più accurata stima complessiva.

3. Le dimensioni del software

Si ritiene da più parti che il fattore che maggiormente influenza i costi del software sia la dimensione dell'applicazione da sviluppare. Si sono storicamente utilizzate misure che intendevano catturare uno dei seguenti due tipi di dimensione.

- Dimensione "interna" (o strutturale) di un'applicazione. Le linee di codice rappresentano l'esponente più caratteristico di questo tipo di dimensione; sono state utilizzate storicamente sigle diverse per denotare misure basate sul concetto di linea di codice o di istruzione (statisticamente molto correlato a quello di linea di codice), tra cui LOC (Lines of Code), KLOC (Kilo Lines of Code), SLOC (Source Lines Of Code), DSLOC (Delivered SLOC), DSI (Delivered Source Instructions), KDSI (Kilo DSI)utilizzate.
- Dimensione "esterna" (o funzionale) dell'applicazione. I Punti Funzione e le loro varianti sono l'esempio principe delle misure di tale categoria [GH01].

Ciascuno dei due tipi di misure di dimensione presenta vantaggi e svantaggi. In particolare, le metriche di dimensione "interna" presentano i seguenti vantaggi:

1. sono facilmente definibili;
2. sono facilmente misurabili;
3. sono facilmente interpretabili;
4. sono utilizzate in molti modelli esistenti per la stima dei costi e per la misura della produttività;
5. sono presenti in pressoché tutti gli archivi storici di dati raccolti in passato.

Le misure di dimensione "interna" presentano tuttavia i seguenti svantaggi:

1. Non vi è una definizione standard per il numero di linee di codice e molte variazioni esistono a seconda di come si contino le linee di dichiarazione, le linee di commento etc.
2. Il valore di LOC dipende dal linguaggio di programmazione utilizzato e dallo stile di programmazione utilizzato da ciascun programmatore. Ciò rende difficile valutare la dimensione di progetti che utilizzano più linguaggi e confrontare la dimensione di progetti che utilizzano linguaggi differenti.
3. Non si tiene comunque conto della diversa complessità e potenza delle istruzioni di uno stesso linguaggio o di linguaggi diversi;
4. Se la produttività è misurata come numero di linee di codice prodotte per unità di sforzo, si può fittiziamente aumentare la produttività semplicemente aggiungendo linee di codice che in realtà non contribuiscono alla funzionalità.
5. Il valore di LOC è difficile da stimare nelle fasi iniziali del ciclo di vita del software. Ciò costituisce un problema molto rilevante per l'utilizzo pratico di molti modelli di stima dei costi, che sono proprio basati sul valore di LOC.
6. Il valore di LOC dà un'enfasi preminente alla fase di codifica che però è solo una delle fasi dello sviluppo del software e non necessariamente la più costosa o importante.

Le misure "esterne" funzionali presentano invece vantaggi soprattutto in riferimento a tutti gli svantaggi delle misure "interne" ad eccezione del primo. Tuttavia, anch'esse presentano alcuni problemi che ora verranno elencati.

1. I Punti Funzione non sono sostenuti da forti motivazioni di carattere teorico. In particolare, l'attributo del software che misurano, la "funzionalità", non è una caratteristica del software per la cui definizione si possa dire vi sia un largo consenso.
2. I Punti Funzione vengono calcolati in base a un insieme di decisioni che possono essere piuttosto soggettive. Alcuni studi mostrano come l'affidabilità del conteggio dei Punti Funzione sia piuttosto bassa, ovvero il numero di Punti Funzione varia notevolmente al variare della persona che si occupa di effettuare la valutazione.
3. Lo schema di pesi usato per il calcolo dei Punti Funzione è soggettivo e non completamente trasparente.
4. Gli ingressi di base utilizzati per il calcolo dei Punti Funzione non sono ortogonali. Inoltre, i pesi forniti fanno sì che vengano persi alcuni gradi di libertà che hanno ad esempio i modelli di regressione, che sono in grado di spiegare in misura maggiore la varianza della variabile dipendente costo.

5. I Punti Funzione sono applicabili al meglio per i Management Information System, ma si prestano meno bene in altri tipi di sistemi.

Alcuni di questi problemi sono stati parzialmente corretti nelle ultime versioni dei Punti Funzione oppure nelle loro varianti, ma il risultato non è ancora del tutto soddisfacente.

Ad ogni modo, alcuni studi indicano che vi è una buona correlazione tra le misure "interne" e quelle "esterne". Il rapporto di proporzionalità tra le due viene a dipendere, come atteso, dal linguaggio utilizzato per la codifica. Ciò ha permesso di derivare dati relativi alle misure di tipo funzionale e tramite un fattore di conversione opportuno ottenere un valore corrispondente di misure "interne" che poi è stato utilizzato all'interno di modelli di costo.

4. Metodi per la stima dei costi del software

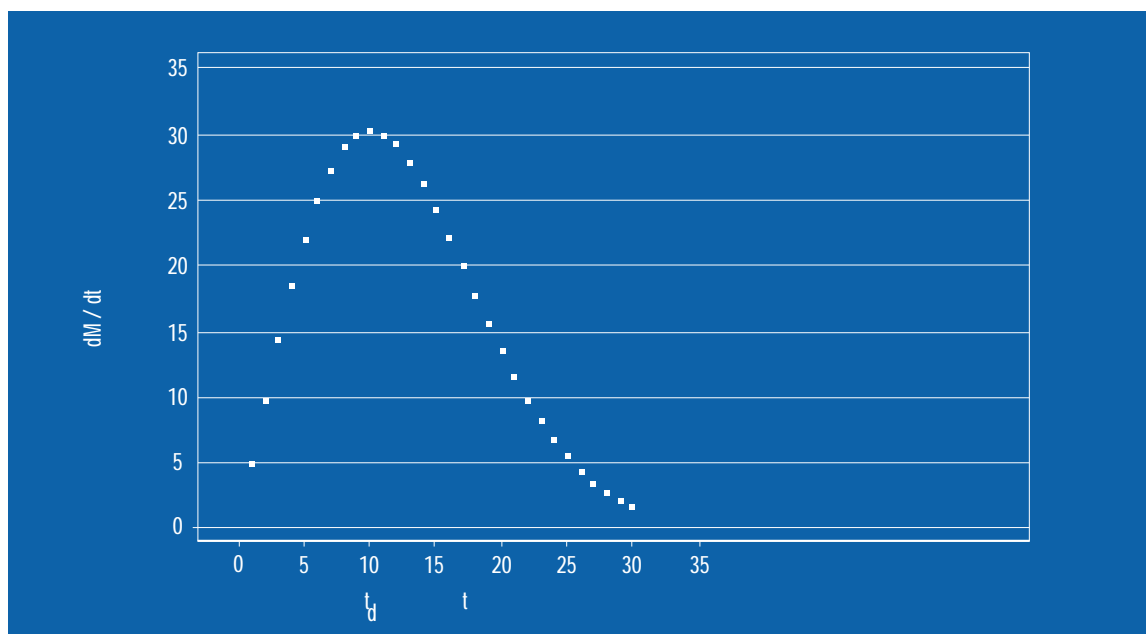
Verranno ora passati in rassegna alcuni metodi utilizzati per la stima dei costi del software, secondo la classificazione esposta nella sezione 2.

4.1 Metodi basati su modelli proprietari

Esistono diversi metodi basati su modelli proprietari. Qui verrà descritto il metodo di Putnam, su cui in parte si basa SLIM. Altri metodi comunque verranno descritti nella parte riguardante il confronto tra i metodi.

Secondo Putnam [P78] la curva di Raleigh-Norden (si veda la figura 2) è in grado di descrivere adeguatamente l'andamento del numero di risorse necessarie nel corso del progetto (ipotesi verificata in circa 200 progetti reali).

Figura 2. La curva di Raleigh-Norden.



La formula che definisce tale curva è

$$dM/dt = K/t_d^2 \cdot t \cdot \exp(-t^2/2t_d^2)$$

nella quale

- t_d rappresenta il mese in cui lo sforzo ha l'intensità massima. Tale momento coincide con la fine della codifica.
- K rappresenta lo sforzo totale in mesi-uomo.

In base a tale formula e dopo una serie di approssimazioni e assunzioni, si ricava che

$$S = C_k \cdot K^{1/3} \cdot t_d^{4/3}$$

dove

- C_k rappresenta una costante che misura lo stato della tecnologia (che sperimentalmente risulta compresa tra 610 a 57314).

Un risultato significativo di tale modello è la relazione tra tempo e sforzo di sviluppo. Se si considera uno specifico prodotto, la dimensione dell'applicazione è una costante, per cui si ricava che

$$K = \text{costante} / t_d^4$$

Tale formula mostra come a modeste variazioni nel tempo di sviluppo possano corrispondere notevoli differenze nello sforzo complessivo necessario, a causa della quarta potenza a cui è elevato t_d nel denominatore dell'espressione a secondo membro. In altre parole, se il tempo di sviluppo (bisogna sottolineare che tale tempo t_d non comprende le attività di verifica e manutenzione) viene accorciato anche di poco, circostanza che si verifica ad esempio quando vi sono scadenze strette da rispettare, lo sforzo necessario si dilata in maniera ben più che apprezzabile. Questo modello è stato il punto di partenza per SLIM, i cui dettagli sono però proprietari.

4.2 Metodi basati su modelli non proprietari

Verrà brevemente esposto il modello COCOMO, che è uno dei più consolidati modelli per la stima dei costi del software.

Il modello COCOMO (CONstructive COSt MOdel, sviluppato da Barry Boehm all'inizio degli anni '80) [B81] assume che lo sviluppo software si svolga secondo il ciclo di vita a cascata, in particolare secondo le quattro fasi seguenti:

1. Pianificazione ed analisi dei requisiti;
2. Progetto dell'applicazione;
3. Sviluppo;
4. Integrazione e test.

Il modello COCOMO è espresso da più formule che forniscono lo sforzo e il tempo necessario per lo svolgimento delle attività dal progetto fino al test di accettazione. I costi per la pianificazione e l'analisi dei requisiti sono calcolati a parte. Tali formule sono state ottenute in base ai dati relativi ai costi di una serie di progetti reali appartenenti ad alcuni settori applicativi.

Il modello COCOMO distingue i tre tipi seguenti di applicazioni:

1. Semplici (Organic Mode): applicazioni relativamente semplici e di limitate dimensioni, che non impongono requisiti particolarmente stringenti (es. tradizionali applicazioni per sistemi informativi).
2. Intermedie (Semi-detached Mode): applicazioni di complessità e dimensioni medie (es. sistema per la gestione di transazioni; in generale, software di base).
3. Complesse (Embedded Mode): applicazioni che richiedono un attento controllo del processo di sviluppo e l'applicazione rigida di precise normative di controllo di qualità (es.: sistemi di controllo per il volo; in generale, sistemi in tempo reale).

Per ciascun tipo di applicazione, l'approccio COCOMO fornisce stime secondo tre modelli a livello di raffinamento crescente, il modello base, il modello intermedio e il modello avanzato.

Nel modello base l'unica variabile indipendente considerata è la dimensione S in KLOC (migliaia di linee di codice) dell'applicazione, che viene utilizzata per calcolare innanzitutto lo sforzo M in mesi-persona e il tempo di sviluppo T in mesi

$$M = a_b S^{b_b} \quad T = c_b M^{d_b}$$

con valori dati per i parametri a_b , b_b , c_b e d_b . Il modello COCOMO base fornisce anche la ripartizione di sforzo e tempo tra le diverse fasi in base a percentuali che dipendono dalle dimensioni dell'applicazione. Inoltre, il modello COCOMO permette di effettuare anche il dimensionamento del team di sviluppo per ciascuna fase. Il numero di appartenenti al team di sviluppo in una qualsiasi fase si ottiene dividendo lo sforzo necessario per la fase per il tempo necessario per svolgere la fase.

Nel COCOMO intermedio viene introdotto un insieme di 15 fattori per le categorie di attributi del prodotto, attributi del calcolatore, attributi del personale, attributi del progetto. Ciascun fattore viene associato a un moltiplicatore il cui valore dipende da caratteristiche e vincoli del progetto. Ad esempio, tra gli attributi del prodotto, la complessità del prodotto viene valutata come molto bassa (moltiplicatore 0,7), bassa (moltiplicatore 0,85), nominale (moltiplicatore 1), alta (moltiplicatore 1,15), molto alta (moltiplicatore 1,3), extra alta (moltiplicatore 1,6).

Pertanto, tali fattori sono associati a una misura ordinale, la definizione dei cui valori (ad esempio la definizione di quando la complessità è alta) sono ampiamente dettagliati in [B81]. Ciascun fattore viene valutato e il moltiplicatore corrispondente (molti) viene selezionato per correggere la stima cosiddetta "nominale" MNom effettuata con formula simile a quella del COCOMO base, seppure con coefficienti diversi

$$M_{\text{Nom}} = a_b S_b^b$$

$$M = M_{\text{Nom}} \prod_{j=1}^{15} \text{molt}_j$$

Il calcolo del tempo di sviluppo si effettua con la formula del COCOMO base.

Non è chiaro se il COCOMO avanzato fornisca risultati di attendibilità tanto superiore a quella del COCOMO intermedio da giustificare in termini di costi e benefici il suo utilizzo.

Nel COCOMO II [B00] vengono apportate modifiche al COCOMO I. Vengono definiti tre livelli di COCOMO, che corrispondono a tre classi di applicazioni e utilizzano informazioni reperibili in tre diversi momenti del ciclo di vita:

dove

- **Application Composition:** è il settore di chi utilizza "primitive" di alto livello per programmare (ad esempio GUI, spreadsheet) o si riferisce alla fase in cui viene generalmente effettuata una prototipazione per risolvere problemi ad alto rischio (interfacce utente, interazione software/sistema, prestazioni, maturità della tecnologia); vengono utilizzati gli Object Points, in quanto si ritiene che siano disponibili in maniera affidabile in questa fase
- **Early Design:** si passa alla fase di progetto, con l'esplorazione di alternative per le architetture; vengono usati i Function Points, il linguaggio e alcuni driver di costo in numero limitato, poiché in questa fase non è ancora noto molto a riguardo del progetto
- **Post-Architecture:** si giunge alla fase di sviluppo tradizionale; si ottiene la stessa precisione di COCOMO 1; vengono usati LOC, Function Points e linguaggio con 17 cost drivers e 5 fattori di scala, che sostituiscono i fattori del modello precedente e i modelli Organic, Semi-detached e Embedded

Inoltre, il modello COCOMO II

- utilizza modelli non lineari per riuso e reingegnerizzazione
- permette il trattamento delle diseconomie di scala
- apporta modifiche ai moltiplicatori originari
- si avvale di un modello Bayesiano per i parametri del modello valutati da esperti.

4.3 Metodi basati su modelli data driven

Sono stati sviluppati molti altri modelli che, a partire da diversi approcci, forniscono strumenti per stimare i costi di sviluppo di una applicazione informatica. Verranno qui mostrate alcune delle formule più rappresentative. È necessario però mettere in guardia dalla possibilità di utilizzare queste formule senza uno studio più approfondito delle condizioni applicative sotto le quali sono state sviluppate. Lo stato dell'arte nell'ingegneria del software empirica non è ancora tale da permettere il riutilizzo di modelli senza che si valuti bene se le condizioni per la loro applicabilità sono soddisfatte.

- 1965 SDC Model: modello di regressione che utilizza 14 parametri [N66]

- Modelli di Doty [HPRS77]:
 - Sforzo = $5.288 \text{ KDSI}^{1.047}$, se $\text{KDSI} \geq 10$
 - Sforzo = $2.06 \text{ KDSI}^{1.047} \cdot \{14 \text{ moltiplicatori}\}$, se $\text{KDSI} < 10$
- Modello di Walston-Felix (IBM) [WF77]:
 - Produttività = somma pesata di 29 cost drivers di processo e prodotto con pesi (basso = -1, medio = 0, alto = 1), dove ogni cost driver è dato da $0.5 \log_{10} \text{PCi}$ (Productivity Change: da basso a alto)
 - Sforzo = $5.2 \text{ KDSI}^{0.91}$
- Modello di Bailey-Basili [BB81]:
 - Sforzo = $5.5 + 0.73 \text{ KDSI}^{1.16}$

Sembra comunque possibile affermare che tali modelli presentano alcune similarità, quale il fatto che l'esponente a cui è elevata la dimensione del codice ha un valore nell'intorno dell'unità.

In generale modelli possono essere costruiti con tecniche statistiche oppure di machine learning, di cui la stima dei costi costituisce un'applicazione tra le tante. Per quanto riguarda le tecniche statistiche, tipicamente vengono utilizzati vari tipi di regressione, quali la regressione lineare, la regressione con curve di ordine superiore, oppure la cosiddetta regressione robusta [RL87], che cerca di alleviare alcuni dei problemi insiti nelle tecniche di regressione "tradizionali", come la presenza di punti fuori scala, che possono far ingiustamente deflettere la retta di regressione, e le ipotesi sulla distribuzione statistica dei residui intorno alla curva di regressione, che deve essere gaussiana.

4.3.1 Alberi di classificazione e Optimized Set Reduction

Gli alberi di classificazione costituiscono la tecnica più usata di machine learning [Q86]. La versione originaria degli alberi di classificazione può essere applicata soltanto con dati di tipo discreto per cui in questa descrizione per il momento si supponrà di avere fattori di costo discreti e il costo come variabile discreta e non continua, per raggruppamento in categorie dei valori possibili di costo. Si procede alla costruzione di un albero per raffinamenti successivi. La radice dell'albero è associata alla totalità dei dati raccolti. Si determina il fattore di costo i cui i valori permettono di predire meglio le categorie di costo. I valori di questo fattore di costo individuano sottoinsiemi dei dati originari e ciascun sottoinsieme può essere rappresentato da un nodo figlio della radice dell'albero. Ciascuno di questi sottoinsiemi avrà una distribuzione dei valori della variabile discreta costo. L'obiettivo è di rendere ciascuna di queste distribuzioni il meno omogenea possibile, ovvero far sì che per quanto possibile in ciascuna di queste distribuzioni ci sia un valore della variabile costo predominante. La valutazione dell'omogeneità di una distribuzione viene effettuata con diversi indicatori, in buona parte basati sulla Teoria dell'Informazione. Se ad esempio ciascuna di queste distribuzioni fosse talmente polarizzata da avere un solo valore, allora si avrebbe una perfetta capacità predittiva del fattore considerato. Tuttavia, tale caso ideale si presenta molto raramente, per cui si procede in modo ricorsivo con l'individuazione di un nuovo fattore per ciascuno dei nodi direttamente dipendenti dalla radice. Per ciascuno di questi nodi vengono perciò individuati sottoinsiemi e creati nuovi nodi figli, cia-

scuno dei quali è individuato da un particolare valore dell'attributo scelto nella generazione dei nodi di primo livello e da un particolare valore dell'attributo scelto per individuarlo a partire dal suo nodo genitore. Il procedimento per raffinamenti successivi si interrompe quando (1) si giunga a un nodo che abbia una distribuzione della variabile costo perfettamente polarizzata, oppure (2) quando il numero delle osservazioni appartenenti al sottoinsieme associato a un nodo scenda sotto un limite predefinito. Così come avviene per i nodi intermedi, anche ogni nodo foglia è individuato da un insieme di valori per i fattori di costo che sono stati utilizzati nel percorso dalla radice alla foglia. Tale insieme di valori è utilizzato per predire la categoria di costo per quel nodo foglia.

Tale procedimento richiede però la discretizzazione di fattori continui, ove presenti, e soprattutto la discretizzazione della variabile costo. Tali discretizzazioni vengono effettuate in maniera soggettiva e possono perciò portare a risultati diversi a seconda del criterio che viene utilizzato. Si sono perciò introdotti i CART (Classification And Regression Tree) [BFOS84], che permettono in qualche misura di ovviare a questo problema, anche se in maniera non del tutto soddisfacente. Un'altra tecnica di machine learning che è stata introdotta sulla base degli alberi di classificazione è OSR (Optimized Set Reduction), in cui viene costruita una gerarchia di nodi che non è necessariamente un albero [BBT92].

4.4 Metodi composti

In tali metodi si combinano modelli analitici e conoscenze degli esperti.

4.4.1 COBRA

COBRA (COSt Estimation Benchmarking and Risk Assessment) è un metodo ibrido per la stima dei costi che combina approcci algoritmici ed empirici [BEB98]. L'idea fondamentale di COBRA è lo sviluppo di un modello di stima della produttività che consiste di due componenti. La prima è un modello causale che genera una stima dell'overhead dei costi, basato su un insieme di fattori di costo locali. Il modello causale viene ottenuto tramite l'acquisizione delle conoscenze di esperti.

La seconda componente di COBRA usa dati da progetti passati ritenuti simili che identificano una relazione tra l'overhead del costo e la produttività. Con COBRA si può valutare l'incertezza di una predizione, si possono effettuare predizioni puntuali per la produttività o lo sforzo, si può effettuare il benchmarking di un progetto e si possono effettuare un'analisi e una valutazione dei rischi.

COBRA assume che i fattori siano ortogonali e che siano misurati con scale quantomeno approssimativamente a intervalli. Non è richiesta nessuna misura specifica della dimensione, ma è comunque necessario valutare la dimensione del sistema. Tra i vantaggi di COBRA ci sono (1) la sua applicabilità in ambienti in cui siano disponibili solo pochi dati, (2) la modellazione esplicita di conoscenze riutilizzabili degli esperti sulla stima dei progetti, (3) la maggior facilità di accettazione da parte degli sviluppatori in quanto i modelli derivano dalla loro esperienza. I principali svantaggi di COBRA consistono (1) nella necessità di avere la disponibilità di esperti e (2) nella difficoltà della raccolta di conoscenze presso gli esperti.

4.4.2 La stima per analogia

L'idea principale è di identificare progetti già completati che siano particolarmente simili a un progetto nuovo. In particolare, sarà necessario selezionare

- le caratteristiche rilevanti dei progetti (ovvero i fattori di costo)
- funzioni adeguate di similarità o distanza
- il numero di progetti analoghi che devono essere presi in considerazione per la predizione.

Sono state definite in letteratura molte funzioni di distanza o similarità, ad esempio basate su distanze di tipo euclideo, in cui i singoli fattori possono anche essere pesati in modo diverso oppure normalizzati, in modo da neutralizzare gli effetti dovuti a scale con unità di misura evidentemente diverse. Inoltre, tali funzioni possono tener conto dei diversi tipi di scala di misura (nominale, ordinale, a intervalli, proporzionale) per i dati che si trovano comunemente nelle basi di dati che contengono dati di progetti. Un esempio di una semplice funzione distanza [SS97] è basato su una distanza euclidea con variabili normalizzate tra 0 e 1. La distanza complessiva distanza(P_i, P_j) tra due progetti P_i e P_j è definita come

$$\text{distanza}(P_i, P_j) = \sqrt{\frac{\sum_{k \in 1..n} \delta^2(P_{ik}, P_{jk})}{n}}$$

dove P_{ik} e P_{jk} sono i valori del k-esimo attributo rispettivamente nel progetto P_i e nel progetto P_j , n è il numero di attributi presi in considerazione, e $d(P_{ik}, P_{jk})$ è dato da

- $\frac{|P_{ik} - P_{jk}|}{\max_k - \min_k}$ se il k-esimo attributo è continuo (\max_k e \min_k sono rispettivamente il valore massimo e il valore minimo possibile per il k-esimo attributo)
- 0 se il k-esimo attributo è categorico e $P_{ik} = P_{jk}$
- 1 se il k-esimo attributo è categorico e $P_{ik} \neq P_{jk}$

La tabella 2 contiene i dati relativi agli attributi di un nuovo progetto e di due progetti già svolti.

Attributi	Nuovo progetto	Progetto 1	Progetto 2
Categoria	Tempo reale	Tempo reale	Simulatori
Linguaggio	C	C	C
Dimensione team	10	10	9
Dimensione sistema	150	200	175
Sforzo	?	1000	950
Similarità		80%	75%

A scopi di predizione si può usare in prima istanza il valore di sforzo del progetto più simile, ovvero 1000 nel caso della tabella 2. In alternativa, se si considerano più progetti ritenuti simili si può utilizzare una media pesata. Inoltre si possono utilizzare meccanismi di aggiustamento basati su criteri oggettivi oppure soggettivi. Come esempio di aggiustamento di tipo oggettivo, dal momento che il nuovo sistema avrà una dimensione che è il 75% della dimensione del Progetto 1, si può sti-

mare che lo sforzo per il nuovo progetto sarà 750, ovvero il 75% dello sforzo per il Progetto 1. Aggiustamenti di tipo soggettivo richiedono le conoscenze di esperti. Ad esempio gli esperti possono ritenere che rispetto al Progetto 1: le funzionalità da realizzare nel progetto presente possono essere quantificate come superiori del 50%, sarà possibile ottenere una riduzione di costi del 15% in quanto parte del team ha già lavorato nel progetto passato e del 10% in quanto parte del codice può essere riutilizzata. Le differenze tra i progetti passati e quello da svolgere possono essere considerate indipendenti oppure concorrenti. Se si considerano indipendenti gli effetti delle differenze, si ottiene il seguente sforzo per il progetto corrente

$$M = 1000 \cdot (1 + 0.5 - 0.15 - 0.1) = 1250$$

Se si considerano concorrenti gli effetti delle differenze, si ottiene il seguente sforzo per il progetto corrente

$$M = 1000 \cdot 1.5 \cdot 0.85 \cdot 0.9 = 1147.5$$

I risultati sono evidentemente fortemente dipendenti dall'esperto scelto. Per ovviare a tale problema si utilizzano metodi basati su gruppi di esperti, quale il metodo Delphi.

4.5 Metodi non basati su modelli

4.5.1 Metodo Delphi

La Rand Corporation ha introdotto il metodo Delphi a partire dal 1948. Il metodo è poi stato utilizzato da altre società in contesti diversi. Un processo di stima basato solo su un esperto è ovviamente fortemente influenzato dalla persona scelta. Per ovviare a tale problema si possono utilizzare gruppi di esperti. Lo scopo del metodo Delphi è di giungere al consenso nel processo decisionale di un gruppo di esperti che sono chiamati a pronunciarsi sulla stima di una grandezza, in questo caso il costo di un progetto software. Verranno qui presentate la versione originale del metodo Delphi e una variante utilizzata per favorirne la convergenza [B81].

Delphi Standard

Gli esperti, sotto la supervisione di un coordinatore, effettuano una successione di stime secondo la procedura seguente:

1. Il coordinatore distribuisce separatamente a ciascun esperto del gruppo la descrizione del problema.
2. Ciascun esperto fornisce la propria risposta in modo anonimo: può consultarsi solo con il coordinatore.
3. Il coordinatore prepara un riassunto delle risposte che include anche le giustificazioni fornite dagli esperti.
4. Il coordinatore invia il riassunto a tutti gli esperti e chiede le ragioni delle differenze tra le stime che hanno fatto e la stima "centrale".
5. Il processo viene iterato per quanto necessario

Delphi Wideband

1. Un coordinatore fornisce a ciascun esperto una specifica e un modulo di stima.
2. Il coordinatore organizza una riunione tra gli esperti.
3. Gli esperti compilano il modulo in maniera anonima.
4. Il coordinatore riceve i moduli e riassume le stime.
5. Il coordinatore invia il riassunto a tutti gli esperti ma non chiede le ragioni delle differenze tra le stime che hanno fatto e la stima "centrale".
6. Il coordinatore organizza una nuova riunione tra gli esperti per discutere delle differenze più marcate.
7. Gli esperti compilano il modulo in maniera anonima.
8. Il processo viene iterato per quanto necessario.

4.5.2 Stima alla Parkinson

L'idea è che un progetto software consumerà tutto il budget a sua disposizione. Caratteristiche:

1. se la stima è in realtà una sovrastima, può portare all'aggiunta di funzionalità solo cosmetiche e al consumo inutile di risorse;
2. prescinde totalmente dall'applicazione, dall'ambiente etc.;
3. generalmente tende a far impiegare metodi di sviluppo scadenti.

4.5.3 Stima Price-to-win

Viene utilizzata per aggiudicarsi una gara o comunque un contratto. Caratteristiche:

- può essere effettivamente utile per aggiudicarsi un gran numero di contratti;
- porta però a molti problemi di processo (ad esempio difficoltà di rispettare i tempi stabiliti nei costi previsti) e il prodotto (ad esempio qualità inferiore);
- può provocare danni di immagine e pertanto di sopravvivenza dell'organizzazione di sviluppo nel lungo termine.

5. Valutazione dei metodi di stima

Verranno qui confrontati i vari metodi di stima dei costi, innanzitutto all'interno delle singole categorie e poi tra una categoria e l'altra.

5.1 Valutazione dei metodi all'interno delle singole categorie

5.1.1 Valutazione dei metodi basati su modelli data driven

Qualità dei modelli e delle stime. Nella maggioranza degli studi empirici effettuati, le tecniche utilizzate (OLS, ANOVA, RR, CART, OSR) hanno dato risultati soddisfacenti. Tuttavia ciò può anche essere dovuto a un fenomeno di selezione degli studi pubblicati.

Interpretabilità. I modelli generati tramite CART oppure OSR sono interpretabili direttamente da non specialisti in statistica. Viceversa, fenomeni di interazione tra diversi fattori oppure la pre-

senza di punti fuori scala possono talvolta rendere i modelli ricavati con tecniche statistiche difficili da interpretare e da usare.

Assunzioni. OLS richiede il soddisfacimento di diverse assunzioni sulla distribuzione dei dati, quali l'omoschedasticità, la normalità delle distribuzioni dei residui, la relazione funzionale lineare tra i fattori e il costo, il fatto che i fattori di costo abbiano scala quantomeno a intervalli. La tecnica ANOVA, è in grado di utilizzare fattori con scale non necessariamente a intervalli, di essere efficace anche con insiemi di dati di dimensione limitata, e di identificare fattori che influiscono su altri fattori. D'altra parte, richiede che la variabile costo abbia una distribuzione gaussiana. Se ciò non dovesse essere vero, è necessario trasformare la variabile costo a questo scopo. CART e OSR sono basate su meno assunzioni di OLS, ma ciò implica anche che richiedono un maggiore numero di dati per giungere a conclusioni affidabili. Sia CART sia OSR richiedono poi la discretizzazione dei dati, che entro certi limiti può influenzare i risultati ottenuti.

Ripetibilità. Tutti i metodi di questo tipo sono altamente ripetibili, una volta che i criteri soggettivi in alcuni di questi (CART e OSR) siano fissati chiaramente.

Complessità. CART e OSR sono i due metodi più nuovi e anche più complessi, in quanto a ogni passo richiedono più selezioni di variabili per giungere a identificare quella che ottimizza la polarizzazione di ogni nodo sottostante.

Automazione. Tutti i metodi di questo tipo possono essere supportati da strumenti automatici.

Trasparenza. Gli algoritmi per tutti i metodi data driven sono noti, per cui tutti questi metodi sono perfettamente trasparenti.

Copertura applicativa. Si tratta di metodi general purpose, ovvero non unicamente mirati alla stima dei costi e perciò possono essere utilizzati in molti e diversi settori applicativi.

Generalizzabilità. Si tratta di metodi general purpose, ovvero non unicamente mirati alla stima dei costi, per cui possono essere utilizzati in molti e diversi contesti applicativi.

5.1.2 Valutazione dei metodi composti

Qualità dei modelli e delle stime. In base a studi preliminari COBRA sembra essere piuttosto promettente dal punto di vista del MMRE. Metodi di stima per analogia hanno avuto la migliore accuratezza produttiva del 60% dei casi ma la peggiore del 30%.

Interpretabilità. La procedura di modellazione di COBRA fornisce un modello causale di impatto dei fattori di costo sul costo e una quantificazione di tale impatto. Tali modelli sono facilmente interpretabili nel contesto di una specifica organizzazione. Un modello basato su stima per analogia consiste di due componenti principali ovvero una misura di similarità e regole di adattamento, per cui è interpretabile solo in termini di quali fattori di costo vengono selezionati dalla funzione di similarità e di quali pesi relativi vengono loro associati. Tuttavia, non esistono criteri comunemente accettati per la definizione di tali funzioni di similarità e dei pesi da attribuire a ciascun fattore di costo.

Assunzioni. COBRA e la stima per analogia possono utilizzare variabili definite a tutti i livelli di misurazione. Non si assume nessuna particolare relazione funzionale tra il costo e i fattori di costo e non vengono fatte assunzioni a riguardo della distribuzione dei dati. COBRA è un mo-

dello additivo, che ha lo scopo di definire fattori di costo ortogonali, sebbene permetta interazioni tra variabili. A differenza di quanto si può credere, la stima per analogia richiede conoscenza di esperti per la derivazione dei modelli, in particolare per definire funzioni di similarità o regole di adattamento appropriate.

Ripetibilità. Nonostante gli algoritmi per la stima per analogia siano ben documentati, il metodo può non essere interamente ripetibile, dal momento che è possibile utilizzare diversi tipi di funzione similarità, di regole di adattamento, oppure modi di usare gli analoghi una volta che siano identificati. Procedure di acquisizione della conoscenza sono parte di COBRA, ma i risultati dipendono comunque dalla selezione degli esperti, che possono anche non essere rappresentativi.

Complessità. La costruzione di un modello COBRA da zero è relativamente complessa poiché riguarda l'acquisizione di conoscenza con l'ausilio di esperti. Sebbene si dica che la stima per analogia sia semplice, è comunque necessario determinare una funzione similarità e regole di adattamento e ciò può non essere semplice viste le insufficienti conoscenze sull'argomento.

Automazione. Esistono diversi strumenti per i metodi analogici. Per COBRA l'automazione non richiede particolari sviluppi di software.

Trasparenza. Entrambi i metodi sono pubblici. Tuttavia, una documentazione migliore in termini di esempi, dettagli e linee guida aiuterebbe una migliore applicabilità di COBRA.

Copertura applicativa. I metodi analogici vengono utilizzati a scopi di predizione e di benchmarking. Le predizioni possono essere ottenute come media dei valori ottenuti dai progetti considerati analoghi, mentre è necessario un numero sufficiente di progetti analoghi per avere una base adeguata per i confronti. Un modello COBRA può essere usato per predizione, benchmarking e valutazione del rischio.

Generalizzabilità. Esiste un corpus di pubblicazioni non esteso su COBRA, mentre metodi analogici sono stati utilizzati in più ambienti.

5.1.3 Valutazione dei metodi basati su modelli non proprietari

SLIM viene trattato in questa sezione, poiché è quantomeno in parte basato su un modello non proprietario

Qualità dei modelli e delle stime. L'accuratezza del metodo di Putnam è stata valutata in diversi studi con risultati insoddisfacenti in molti casi. In particolare in studi comparativi il metodo di Putnam ha dato risultati decisamente poco incoraggianti nell'80% dei casi.

La convalida iniziale del COCOMO fu effettuata su dati provenienti dalla TRW. Vennero ottenuti ottimi risultati per il COCOMO intermedio e il COCOMO avanzato. Valutazioni indipendenti tuttavia non hanno sempre portato a risultati altrettanto positivi. L'accuratezza in termini di MMRE varia tra molto buona ($MMRE < 0,25$) e pessima ($MMRE = 7,58$). Alcuni studi hanno riscontrato che il COCOMO intermedio e avanzato non producono necessariamente risultati migliori e che il modello tende a sovrastimare sistematicamente lo sforzo. In altri studi si è invece trovato che il COCOMO tende a sottostimare lo sforzo. Inoltre i modelli intermedio e avanzato hanno anche problemi di overfitting. Non esistono ancora studi particolarmente convincenti a ri-

guardo delle prestazioni del COCOMO II, soprattutto in confronto ad altri metodi esistenti.

Input richiesti. L'uso del metodo SLIM richiede la stima della dimensione del sistema, la determinazione del fattore tecnologico e valori appropriati per l'accelerazione della manodopera. Il fattore tecnologico e l'accelerazione della manodopera possono essere calcolati sulla base dei progetti passati con caratteristiche simili. La dimensione del sistema in termini di KDSI deve essere stimata in maniera soggettiva. Questo è chiaramente uno svantaggio a causa della difficoltà nella stima delle KDSI all'inizio di un progetto e dalla dipendenza di tale misura dal linguaggio di programmazione.

Il COCOMO base richiede che l'utente fornisca il numero di KDSI come misura di dimensione per determinare una stima iniziale dello sforzo. Per i modelli intermedio e avanzato è richiesta anche la valutazione di 15 fattori di costo generici e poi diversi parametri riguardanti il riuso del codice. La valutazione di molti di questi fattori di costo e parametri può essere altamente soggettiva. Il COCOMO II lavora sulla base di diverse misure di dimensione (FP, LOC, Object Points). Un numero diverso di fattori di costo e di misure di dimensione è richiesto per ciascun tipo di modello.

Completezza delle stime. Il modello SLIM fornisce stime per lo sforzo, la durata e l'allocazione delle risorse umane per il ciclo di vita del software. Il COCOMO I fornisce equazioni per stimare lo sforzo, la durata che gestisce l'effetto del riutilizzo del codice da altri progetti. Il COCOMO II fornisce stime di costo, sforzo e disposizione temporale delle attività (schedule), a seconda del modello utilizzato. Gestisce anche l'effetto del riuso, della reingegnerizzazione e della manutenzione.

Tipo di stima. Il metodo di Putnam fornisce principalmente stime puntuali, così come il COCOMO I. Il COCOMO II fornisce una stima puntuale e un intervallo di incertezza che dipende dalla completezza degli ingressi forniti.

Interpretabilità. Tutti questi modelli sono relativamente difficili da interpretare per gli sviluppatori, a causa della loro struttura. Le equazioni funzionali devono essere comprese e i coefficienti devono essere interpretati. Inoltre non vi è nessuna garanzia che questi modelli generici rappresentino adeguatamente la realtà di un'organizzazione.

Assunzioni. SLIM suppone che la forza lavoro vari secondo la curva di Raleigh, che in realtà non modella adeguatamente ciò che avviene per progetti di dimensioni piccole e medie. Il COCOMO suppone che vi sia una relazione di tipo polinomiale tra la dimensione e lo sforzo. I valori degli esponenti mostrano come si ritenga che vi siano necessariamente diseconomie di scala. Tale assunzione è stata dimostrata inadeguata in alcuni studi. COCOMO I assume che il costo del riutilizzo sia una funzione lineare del grado di modifica del software riutilizzato. COCOMO II usa un modello di stima non lineare, sulla base dei risultati ottenuti tramite l'analisi di dati provenienti dalla NASA.

Ripetibilità. Sebbene gli algoritmi siano ben documentati, la loro ripetibilità dipende dalla soggettività nella stima dei parametri di ingresso richiesti, che può essere influenzata dall'esperienza di chi sviluppa il modello, dallo strumento utilizzato per ottenere le stime e dal momento in cui viene effettuata la stima. Tuttavia il COCOMO II tiene conto dell'incertezza nelle informazioni d'ingresso.

Complessità. Il modello SLIM ha complessità relativamente bassa. La complessità del COCOMO aumenta all'aumentare del livello di dettaglio del modello.

Automazione. Sia il metodo di Putnam sia il COCOMO sono supportati da strumenti automatizzati.

Trasparenza. SLIM ha un metodo proprietario, mentre il COCOMO è pubblicamente disponibile.

Copertura applicativa. SLIM mira a investigare relazioni tra i livelli di manodopera, lo scheduling e lo sforzo. Lo strumento che lo supporta permette di investigare i trade-off tra i fattori di costo e gli effetti dell'incertezza nelle stime di dimensione. COCOMO I non fornisce modalità per la gestione dei rischi. COCOMO II fornisce un intervallo di stima invece che una mera stima puntuale e permette perciò di avere una base per la stima del rischio.

Generalizzabilità. Il modello SLIM ha l'ambizione di essere valido in generale per grandi sistemi. Il COCOMO I fu sviluppato all'interno di un processo tradizionale a cascata, per cui poteva non essere applicabile ad altri tipi di sviluppo. Il COCOMO II è stato adattato per soddisfare le necessità di nuove pratiche di sviluppo.

Granularità. I modelli COCOMO I e COCOMO II forniscono un livello di granularità sicuramente superiore rispetto a SLIM, in quanto forniscono distribuzioni dettagliate per attività su sforzo e scheduling.

Disponibilità delle stime. I modelli SLIM e COCOMO I non possono in realtà fornire stime affidabili nelle prime fasi del ciclo di vita poiché richiedono una stima affidabile della dimensione dell'applicazione. Il modello COCOMO II fornisce stime a livelli diversi di granularità a seconda dello stadio di sviluppo.

5.1.4 Valutazione dei metodi proprietari

Sebbene non sia possibile descrivere questi metodi in dettaglio dal momento che il loro dettaglio interno non sono disponibili pubblicamente, verrà qui presentata una valutazione basata sulle informazioni che sono disponibili. SLIM, di cui una parte dei dettagli è nota, è presentato tra i metodi non proprietari.

Qualità del modello e delle stime. Esistono pochissimi studi indipendenti che permettano di confrontare questi metodi. In uno studio del 1987 ESTIMACS ha dato risultati migliori di SLIM, COCOMO e OLS.

Input richiesti. ESTIMACS usa una misura simile ai Punti Funzione. PRICE-S richiede la conoscenza della dimensione del progetto, l'area applicativa del progetto, il livello di progettazione di codifica da effettuare ex novo e altre informazioni. La stima della dimensione del sistema è ottenuta tramite un modulo specifico che permette di ricavare stime di Punti Funzione, LOC e varianti object-oriented dei Feature Points.

Completezza della stima. ESTIMACS fornisce risultati riguardanti lo sforzo, lo scheduling, il livello massimo di manodopera, una suddivisione riguardante tali caratteristiche e altre informazioni. PRICE-S fornisce un riassunto dettagliato per mese dello scheduling, dei costi e della manodopera, nonché proiezioni riguardanti il rischio. Knowledge Plan fornisce stime dello sforzo, dello scheduling, del costo, dei requisiti sulle risorse e del livello di qualità.

Tipo di stima. ESTIMACS fornisce stime puntuali e limiti inferiori e superiori. Knowledge Plan fornisce stime puntuali e un livello di accuratezza delle stime, basato sul numero di progetti simili trovati nel database interno.

Interpretabilità. Tale caratteristica non può essere valutata poiché i modelli alla base di questi metodi sono proprietari.

Assunzioni. PRICE-S è un metodo parametrico di stima dei costi, e non è basato su una singola relazione di stima dei costi né su un solo database. Knowledge Plan è un metodo di stima basato sulla conoscenza che era ricavata a partire da 6700 progetti provenienti da molti e svariati ambienti.

Ripetibilità. La ripetibilità dei risultati dipende chiaramente dalle informazioni l'ingresso fornite che permettono di selezionare informazioni diverse appartenenti ai database contenuti negli strumenti.

Complessità. Sulla base del numero delle ingressi richiesti Knowledge Plan sembra essere il metodo più complesso, con PRICE-S ed ESTIMACS al secondo e al terzo posto rispettivamente.

Automazione. Tutti questi metodi sono pienamente supportati da strumenti.

Trasparenza. Né il tipo di modellizzazione né i dettagli implementativi sono public domain.

Copertura applicativa. PRICE-S riesce a tradurre l'incertezza sulle caratteristiche del progetto in una valutazione del rischio relativo ai costi e al non riuscire a rispettare le scadenze. Knowledge Plan supporta la gestione di progetto, la modifica delle stime, mostra l'impatto del progredire del progetto su scheduling, sforzo, costo e qualità, e contiene un'analisi what-if per esplorare strategie alternative a riguardo del personale, del processo e della tecnologia.

Generalizzabilità. In base a informazioni commerciali PRICE-S dovrebbe coprire tutti i tipi di sistema. Knowledge Plan dovrebbe coprire tutti gli ambienti principali di applicazione.

Tuttavia questa informazione non sono confermate da nessuno studio indipendente e pubblicamente disponibile.

Granularità. PRICE-S deriva stime di scheduling e di sforzo per le fasi principali di sviluppo. Inoltre fornisce stime per l'analisi dei requisiti, l'integrazione sw/hw, la manutenzione in tre categorie.

Disponibilità delle stime. PRICE-S e Knowledge Plan forniscono stime iniziali all'inizio del ciclo di vita che possono e dovrebbero essere modificate al procedure del progetto.

5.1.5 Valutazione dei metodi non basati su modelli

Qualità delle stime. Metodi non basati su modelli raggiungono un'alta accuratezza delle stime se sono accoppiati a strumenti o combinati con altri modelli e metodi di stima dei costi. Se usati da soli possono portare a tradizioni molto inaccurate in quanto fortemente dipendenti dal giudizio individuale. Inoltre vi è una tendenza a sottostimare lo sforzo richiesto per un'attività da parte di chi compirà l'attività stessa, come dimostrano diversi studi.

Assunzioni. Non vi sono assunzioni particolari riguardanti i dati provenienti da progetti precedenti. Tuttavia si suppone che gli esperti siano in grado di fornire stime non deviate e accurate. Ciò può essere favorito dall'impiego di tecniche rigorose di raccolta delle informazioni.

Ripetibilità. La ripetibilità di questi metodi viene favorita dall'uso di un processo organizzato, nonostante il procedimento sia largamente soggettivo.

Complessità. La complessità dipende dal numero di esperti utilizzati, dal numero e dall'organizzazione delle riunioni e dal processo da seguire.

Trasparenza. La trasparenza dipende dalla definizione delle tecniche di stima utilizzate. Non è spesso chiaro come ciascun esperto derivi la sua stima.

Generalizzabilità. Il giudizio di esperti è stato largamente usato nella pratica ma scarsamente studiato nell'ingegneria del software.

5.2 Valutazione di tipi di metodi diversi

5.2.1 Metodi basati su modelli e metodi non basati su modelli

I metodi basati su modelli al contrario di quelli non basati su modelli possono essere considerati come conoscenze riutilizzabili. Il processo di stima delle risorse è più indipendente dalle abilità individuali. Inoltre i metodi non basati su modelli richiedono un forte coinvolgimento degli esperti per generare una stima per ogni nuovo progetto. Ciò rende il metodo difficile da applicare poiché in molte organizzazioni gli esperti sono molto richiesti e utilizzati in altre attività. Infine non è possibile ridurre il costo necessario per derivare una stima e migliorare l'accuratezza da un progetto all'altro poiché non vi è nessun processo organizzato di accrescimento delle conoscenze.

5.2.2 Metodi basati su modelli generici e metodi basati su modelli specifici

I modelli generici partono da una relazione che si ritiene universalmente valida tra le risorse e la dimensione del sistema. In realtà sembrerebbe ragionevole supporre che in ambienti diversi vi siano relazioni diverse. Quando si utilizzano metodi basati su modelli specifici è necessario investigare le relazioni sulla base dei dati raccolti.

Algoritmi generici possono produrre risultati altamente inaccurati se vengono usati senza essere calibrati in contesti diversi. La ragione è che usano fattori di costo predefiniti che non sono necessariamente validi in tutti i contesti.

5.2.3 Metodi proprietari e metodi non proprietari

I modelli proprietari sono realizzati come scatole nere e perciò l'algoritmo utilizzato non è noto all'utente. E perciò difficile giustificare una stima generata con un metodo proprietario e la sua credibilità è basata unicamente sulla credibilità del costruttore dello strumento. Pertanto, i metodi basati su modelli proprietari hanno più difficoltà a essere accettati.

5.2.4 Metodi basati su modelli data driven e metodi composti

Negli studi empirici nell'ingegneria del software uno dei problemi fondamentali è la relativa scarsità di osservazioni disponibili ovvero i dati che vengono ottenuti in un esperimento sono solitamente pochi se confrontati a quello che avviene in altre discipline. Inoltre di insieme dei dati è spesso non bilanciato e può avere valori mancanti in alcune osservazioni oppure punti fuori scala. perciò delle conclusioni ricavate unicamente in base ai dati di progetto possono essere fuorvianti oppure difficile da ottenere. È perciò importante dal punto di vista pratico consultare esperti. I metodi composti cercano di incorporare le conoscenze degli esperti all'interno a del processo di costruzione dei modelli. Le conoscenze degli esperti vengono rese esplicite nel modello e ciò ne favorisce il grado di accettabilità.

6. Conclusioni e sviluppi futuri

Il problema della stima dei costi del software è di grande importanza economica, sia per i costruttori sia per gli acquirenti di software. La difficoltà nasce principalmente dal fatto che lo sviluppo del

software è basato soprattutto sulle risorse umane. Inoltre, i progetti software tendono a diversificarsi l'uno dall'altro, per cui non sembra possibile che si possa arrivare a leggi di grande affidabilità e precisione matematica in questo settore. Tuttavia, vi è un corpus di conoscenze iniziale che permetterà in futuro di estendere le proposte esistenti anche a tipologie di applicazioni diverse da quelle per cui la stima dei costi era stata pensata inizialmente. Le applicazioni web costituiscono probabilmente il nuovo settore applicativo più importante a questo scopo [BMP03].

7. Bibliografia

- [B00] B. Boehm et al., "Software Cost Estimation with COCOMO II", Prentice-Hall, 2000.
- [B81] B. Boehm, "Software Engineering Economics", Prentice-Hall, 1981.
- [BB81] J. Bailey, V. Basili, "A Meta-model for Software Development Resource Expenditures", ICSE, IEEE CS Press, pagg. 107-116, 1981.
- [BBT92] L. Briand, V. Basili, W. Thomas, "A Pattern Recognition Approach for Software Engineering Data Analysis", IEEE Transactions on Software Engineering, pagg. 931-942, novembre 1992.
- [BEB98] L. Briand, K. El Emam, F. Bomarius, "COBRA: A Hybrid Method for Project Effort Using Analogies", ICSE 1998, pagg. 390-399, maggio 1998.
- [BFOS84] L. Breiman, J. Friedman, R. Ohlsen, C. Stone, "Classification and Regression Trees", Wadsworth and Brooks/Cole Advanced Books and Software, 1984.
- [BMP03] L. Baresi, S. Morasca, P. Paolini, "Estimating the Design Effort of Web Applications", METRICS '03, pagg. 239-249, settembre 2003.
- [BW00] L. Briand, I. Wiczorek, "Resource Estimation in Software Engineering", ISERN Report n. 2000-05, 2000.
- [GH01] D. Garmus, D. Herron, "Function Point Analysis: Measurement Practices for Successful Software Projects", Addison Wesley, 2001.
- [HPRS77] J. Herd, J. N. Postak, W. E. Russell, K. R. Stewart, Software Cost Estimation Study—Study Results", Technical Report, RADC-TR-77-220, Vol. I, Doty Associates, Inc., Rockville, MD giugno 1977.
- [N66] E. A. Nelson, "Management Handbook for the Estimation of Computer Programming Costs", AD-A648750, System Development Corporation, 1966.
- [P78] L. H. Putnam, "A General Solution to the Macro Software Sizing and Estimating Problem", IEEE Transactions on Software Engineering, Vol. SE-4, No. 4, pagg. 345-361, luglio 1978.
- [Q86] J.R. Quinlan, "Induction of Decision Trees", Machine Learning, vol. 1, n. 1, pagg. 81-106, 1986.
- [RL87] P.J. Rousseeuw, A.M. Leroy, "Robust Regression and Outlier Detection", Wiley Series in Probability and Mathematical Statistics, John Wiley and Sons, 1987.
- [SS97] M. Shepperd, C. Schofield, "Estimating Software Cost Estimation, Benchmarking, and Risk Assessment", IEEE Transactions on Software Engineering, pagg. 736-743, dicembre 1997.
- [WF77] C. E. Walston, C. P. Felix, "A Method of Programming Measurement and Estimation", IBM System Journal, vol. 16, pagg. 54-73, 1977.

6. Tavola rotonda: il punto di vista delle imprese Ict e degli utenti

A chiusura del convegno si è svolta una tavola rotonda, moderatore Franco Patini, vicepresidente Federcomin e presidente Anasin, alla quale hanno partecipato rappresentanti del mondo dell'industria, del settore bancario, di organizzazioni di utenti e di analisti di mercato. Sono intervenuti: Gabriele Ruffatti di Engineering - Ingegneria Informatica Spa, Paolo Ferrara di Etnoteam Spa, Andrea Grezzi di Getronics Spa, Antonio L. Rassa di C&M Spa, Roberto Meli di Gufpi, Romano Stasi dell'Abi, Associazione bancaria italiana, e Annalisa Erba di Gartner.

Pur nella diversità di vedute e di ruoli, il confronto ha messo in evidenza come alcuni punti siano ampiamente condivisi. In particolare:

- maggiore attenzione all'uso di metriche nei progetti della pubblica amministrazione (uso di metriche quali i punti funzione non solo per definire un prezzo "a corpo", ma anche per esprimere livelli di qualità del prodotto); attenzione inoltre a usare correttamente i riferimenti di mercato e i dati delle attività di benchmarking;
- necessità di una maggiore diffusione delle competenze (vale a dire necessità di formazione specifica, ma anche impulso a rendere maggiormente comprensibile e usabile il metodo);
- necessità di evoluzione e di adattamento della metrica dei punti funzione a una realtà mutevole quale il mondo dello sviluppo software.

Abstract dell'intervento di Gabriele Ruffatti - Engineering Ingegneria Informatica

Il tema delle “metriche” nell’ambito dello sviluppo software è estremamente complesso, spesso relegato nell’ambito accademico o della ricerca, e trova una non completa applicazione pratica nell’ambito dell’industria del software.

Su questo, la Pubblica Amministrazione può promuovere una ulteriore standardizzazione, come evoluzione dell’utilizzo della tecnica dei Function Points, al fine di cogliere l’obiettivo primario della realizzazione di progetti informatici che siano pienamente utilizzati, ed ottenere alcuni benefici effetti quali:

- l’aumento della soddisfazione del Cittadino, in quanto Utente di applicazioni efficaci e fruitore di un impiego “etico” del denaro pubblico;
- la diminuzione del gap esistente tra la soddisfazione dell’Amministrazione-Committente e l’esigenza di “giusta remunerazione” delle Aziende Fornitrici;
- l’instaurazione di un circolo virtuoso (Cittadino-Amministrazione-Fornitore) che stimoli il mercato nell’individuazione di progetti pienamente efficaci-efficienti-remunerativi.

Per raggiungere tale obiettivo è necessario individuare regole che siano:

- “certe”, da utilizzare per l’assegnazione dei contratti di fornitura e la valutazione di conformità degli stessi;
- adeguate alla disciplina dello sviluppo software, e che superino le “incertezze” che derivano dalla loro adozione.

La tecnica dei Function Points costituisce un punto di partenza e la sua evoluzione prevista non risolve completamente il problema.

I mutati contesti, sia tecnologico (ad esempio le applicazioni web, datawarehouse, business intelligence), sia di mercato (riduzione dei costi e dei tempi di realizzazione) e la rapida evoluzione dei requisiti delle applicazioni, inducono la necessità di misurare lo sforzo di realizzazione, sia in sede di offerta, che soprattutto in sede di esecuzione e di validazione finale, non solo in termini dimensionali, ma anche in termini qualitativi, intesi come misura del processo produttivo ed, eventualmente, come qualità del prodotto o servizio fornito.

La strada da percorrere evidenzia diverse azioni che possono essere intraprese, tra le quali: - la necessità di inserire nei capitolati in modo esplicito il concetto di definizione del processo produttivo e di adattabilità dello stesso alle modifiche in corso d’opera; - l’opportunità di incrementare le interazioni Amministrazione-Committente/Fornitore in sede di esecuzione; - la separazione delle metriche ad approccio quantitativo (di prodotto) da quelle ad approccio qualitativo, queste ultime in relazione al processo produttivo; - l’opportunità di far evolvere la valutazione della qualificazione dei fornitori verso un criterio di verifica di certificazione in rispondenza all’adozione di processi di sviluppo misurabili (es.: ISO 9001-2000, SW-CMM).

Il senso della misura

La necessità così come l'efficacia dell'utilizzo di metriche di misura dello sforzo di realizzazione del software come uno dei molti parametri di definizione/controllo di importanti contratti di fornitura, è ampiamente dimostrato da importanti esperienze, anche nel settore privato.

Il metodo Function Point in particolare ha dimostrato la propria robustezza e applicabilità nell'ambito dello sviluppo di sistemi "tradizionali".

È però altrettanto evidente che l'efficacia pratica dell'utilizzo di simili metriche come strumenti di definizione e di controllo di importanti contratti di fornitura richiede la capacità di stabilire un dialogo costruttivo e maturo tra il Committente ed il Fornitore, sia dal punto di vista tecnico che da quello gestionale.

Anni e anni di ingegneria di SLA basati su metriche oggettive ci hanno insegnato che le penali non risolvono il problema.

Anche attraverso la crisi degli ultimi anni, è risultato sempre più evidente che le aziende di progettazione e maintenance di "large software systems", che possono garantire risultati di qualità a costi ragionevoli, sono quelle che soddisfano le sette caratteristiche già evidenziate da Capers Jones[1] ormai una decina d'anni fa:

1. They measure software productivity and quality accurately.
2. They plan and estimate software projects accurately.
3. They have capable management and technical staffs.
4. They have good organization structures
5. They have effective software methods and tools.
6. They have adequate staff office environments.
7. They are able to reuse substantial volumes of software deliverables.

Capers Jones sostiene che tutte le 7 caratteristiche sono egualmente importanti, ed aggiunge che senza la prima non si riesce a mettere in pratica le altre.

In altre parole, se sono capace di misurare qualità e produttività del mio apparato di produzione e integrazione di sistemi software, allora posso anche tenere davvero sotto controllo i costi (pianificando efficacemente e "facendo riuso controllato"), valutare la capacità e la produttività delle mie persone e farle crescere insieme alla mia organizzazione (così da essere in grado di ridurre ulteriormente i costi). In definitiva è possibile mettere in pratica tutto ciò che un committente sogna dei propri fornitori. Chi è in grado di misurarsi davvero può sostenere processi maturi.

Capacità di misurarsi e capacità di deliberare sono perciò intimamente connesse. Da un punto di vista speculare, infatti, è altresì vero che l'usabilità e l'effettiva comparabilità dei dati di misura dipende sostanzialmente dalla maturità dei processi di controllo/produzione dell'organizzazione; ma oggi, il vasto insieme dei fornitori della PAL/PAC ha inevitabilmente conseguito caratteristiche di forte disomogeneità anche all'interno delle stesse tipologie di fornitura. Credo quindi nella reale efficacia dell'utilizzo di metriche come effettivo driver contrattuale di controllo del Fornitore, e ciò presuppone che tra le tante cose che il Committente pubblico dovrebbe richiedere al Fornitore dovrebbero avere particolare rilievo prerequisiti specifici di maturità da accertare mediante meccanismi di vendor assessment/rating.

Abstracts dell'intervento di Andrea Grezzi - Getronics

Il tema della misurazione delle prestazioni nell'area di sviluppo e manutenzione del software e, in termini più ampi dei progetti applicativi, è da sempre all'attenzione di Getronics.

D'altro canto, gli Enti della Pubblica Amministrazione hanno avviato da tempo una attività di ricerca e definizione di modelli orientati a misurare e verificare tutti gli indicatori che consentono il governo dei risultati attesi in questi ambiti. Questa realtà di mercato ha favorito, tra i player di riferimento Getronics, che ha messo a punto modalità rigorose e deterministiche per la quantificazione dei risultati attesi e per la corretta determinazione degli impegni necessari, al fine di poter giungere alla proposizione di offerte che coniughino al meglio gli aspetti di qualità e di convenienza, ottimizzando quindi il rapporto price/performance della soluzione proposta.

Getronics, per garantire livelli di eccellenza nello sviluppo applicativo e nell'erogazione di servizi di manutenzione applicativa, adotta un duplice percorso che prevede:

.l'applicazione rigorosa delle tecniche di misurazione dei Function Point in ottemperanza alle indicazioni degli Istituti che ne disciplinano le regole, seguendone con attenzione gli sviluppi conseguenti alla rapida evoluzione delle tecnologie abilitanti;

.il costante confronto dei dati preventivi e di quelli consuntivi relativi ai progetti, con le serie storiche attinte dal proprio Experience Metrics Data Base, dove sono collezionati i dati metrici relativi a molteplici progetti, servizi e consulenze effettuati negli anni per i propri Clienti.

La motivazione della scelta della metrica dei FP da parte di Getronics come elemento base nella propria attività di dimensionamento funzionale, è principalmente dovuta all'esistenza di Gruppi di sviluppo e aggiornamento degli standard d'uso della metrica, che nel tempo si sono estesi dall'ambito strettamente gestionale, prevalentemente main-frame oriented, verso l'applicabilità in ambienti eterogenei.

In particolare tale metrica si è dimostrata valida, nella nostra esperienza, anche per gli sviluppi in ambienti WEB e in personalizzazioni di pacchetti quali il SAP. Attualmente ne stiamo sperimentando l'uso in realtà dove la componente architeturale e di integrazione applicativa è preponderante e dove l'elemento di valutazione della produttività rappresenta un elemento chiave per una corretta quantificazione dei costi di un sistema. In tale ambito operiamo anche con il GUFPI per la ricerca di soluzioni utilizzabili a livello nazionale e internazionale.

Sulle metriche

Nell'ambito dell'affidamento di commesse di sviluppo/manutenzione del software le domande che ci si può porre sono?

Quali metriche? Per quali misure? Con quali mezzi?

Ma prima ancora ci chiediamo: che cosa dobbiamo misurare?

1. La quantità di software (prodotto o mantenuto)
2. La qualità del software
3. La produttività media (per unità di prodotto con data qualità)
4. Livello di costo (per unità di prodotto con data qualità)

La misura della quantità di software

Le tecniche principalmente in uso sono quelle che utilizzano i Function Point e i LOC (line of code). Sono però anche utilizzati metodi che quantificano in base al numero dei moduli (programmi, sottoprogrammi, oggetti, ecc.).

Nel caso di nuovi sviluppi sta assumendo sempre più rilevanza la tecnica basata sui Function Point, che consente peraltro di "stabilizzare", anche ai fini contrattuali, il parametro quantità.

E' bene però precisare che, ai fini della preventivazione economica complessiva, il loro utilizzo può e spesso deve essere accompagnato da altri criteri, quale quello basato sul numero dei moduli. Infatti, dovendo poi procedere a valorizzazioni che tengono conto della produttività media e del livello di costo e considerando che questi parametri sono attendibili solo se derivano da serie storiche rilevate in contesti sufficientemente omogenei (stessa tecnologia, stessi tools, stessi standard, ecc.) o in "evoluzione controllata" e in lassi temporali non inferiori a un paio di anni, può essere molto utile comparare i risultati derivanti da diverse misurazioni.

Per le manutenzioni invece bisogna distinguere fra il caso in cui si sviluppano nuovi moduli in sostituzione o in aggiunta ai precedenti -nel qual caso si ricade nel punto precedente- e il caso in cui la modifica riguarda parti precedentemente codificate di cui si modificano istruzioni (in variazione, aggiunta, eliminazione). In quest'ultimo caso, le valorizzazioni fatte a preventivo (sulla base del numero di Function Point o dei moduli toccati e del peso degli interventi) e convenute contrattualmente, possono essere comparate con valutazioni a posteriori basate sul numero di LOC effettivamente modificate, con lo scopo di valutare e affinare l'affidabilità dei processi di misura a preventivo. Questo può anche produrre un giudizio sui fornitori utile ai fini di future attività.

La misura della qualità del software

Prendendo come riferimento la norma ISO9126 iniziamo a fare alcune esclusioni che, per quanto importanti, riguardano essenzialmente politiche che andrebbero definite a monte della realizzazione e di cui è solo necessario accertarne il rispetto. Le caratteristiche escluse, in linea di massima, sono:

- Portabilità

- Usabilità
- Efficienza

Ciò vale anche per la caratteristica Funzionalità con eccezione della sottocaratteristica Idoneità (cioè funzionalità adeguate ai compiti) che è invece da valutare se l'attività di specifica funzionale rientra o meno nella commessa. E' suggeribile che le specifiche funzionali siano definite a monte, almeno nella misura necessaria a costituire un capitolato tecnico stabile, magari in una commessa a se stante. Ad ogni modo le metriche qualitative possono essere identificate, ma non sono facilmente riutilizzabili perché parte significativa delle rilevazioni avvengono dopo la fine del progetto e comunque hanno scarsa uniformità fra i vari progetti.

Rimangono due caratteristiche: Affidabilità e Mantenibilità.

L'Affidabilità può essere misurata con varie metriche. Per esempio la misura può essere numero di difetti per unità di prodotto (es. ogni 100 Function Point) in un dato periodo di tempo o in una certa fase (es. accettazione). Sono possibili anche altre metriche, che tra l'altro tendono a prevenire la difettosità, e che si basano sulla misura della qualità del test in termini di copertura topologica o funzionale (per es. numero di LOC o di funzionalità collaudate almeno una volta). Naturalmente, in questi casi ma anche negli altri, è preferibile identificare strumenti di misura automatici.

La misura della Mantenibilità può essere basata essenzialmente su ispezione del codice e la determinazione di indici noti quale la Complessità Ciclomatica, ma anche su valutazioni quali il rispetto degli standard (uso di istruzioni o costrutti vietati). Può esser misurata anche la documentazione associata. Fra queste dovrebbe essere considerata anche la documentazione delle prove, in modo da permettere il riuso senza che le stesse debbano essere riprogettate (attività di norma molto costosa). Se si misura la qualità del software su commesse di manutenzione sarà anche necessario definire misure circa la regressione.

La produttività media

Deve essere comunque dominata, tipicamente disponendo di serie storiche per piattaforma di sviluppo. Questo consentirà di prevedere meglio fin dall'inizio e di evitare fenomeni (per esempio dumping) e imprevisti indesiderati (per esempio impossibilità a completare la commessa).

Livello di costo

Esaminiamo sinteticamente da quest'ultimo punto che potrebbe sembrare fuori luogo, ma ritengo che non sia da trascurare, perché fa correre il rischio di perdere di vista alcuni principi cardine dell'organizzazione del lavoro. Questi stabiliscono che è tanto più facile ed economico produrre quanto più è semplice apprendere e mettere in pratica le tecniche di produzione.

Provo a chiarire con qualche esempio. Poniamo che, stabilita una certa tecnologia, un diplomato dopo tre mesi di formazione/affiancamento sia in grado di produrre con il 90% della efficienza prevista. Poniamo invece che, su altra piattaforma tecnologica, un laureato raggiunga il 90% di efficienza dopo un anno. Il risultato sarà che il bacino delle risorse "skillate" sarà molto meno ampio e flessibile nel secondo caso, con conseguente innalzamento di costi diretti e indiretti sulle produzioni ma anche con costi sociali (meno gente lavora).

Naturalmente la tematica è ampia e complessa, che non può essere trascurata e che deve perciò diventare maggiormente oggetto di valutazione e di dibattito.

