

# LA GESTIONE QUANTITATIVA DI PROGETTI SOFTWARE

DOMENICO CLERICI  
LUIGI PETRUZZELLI



## Dalla IV di copertina...

Gli autori del libro si sono proposti di fornire un'introduzione approfondita alla gestione quantitativa di progetti di sviluppo software. Nel contempo si sono riallacciati a quanto con essi è collegato: da un lato i processi, dall'altro la misurazione del software.

Partendo da una panoramica sulla gestione dei progetti e sui cicli di vita, si passa ai punti tradizionali in cui i metodi quantitativi svolgono un ruolo preponderante: stime, rischi, modelli. Si prosegue con una presentazione dei processi aziendali e del loro miglioramento, nonché dei principali standard a essi collegati per quanto concerne il software. Si descrivono poi le metriche: di processo, progetto, prodotto.

Un capitolo è dedicato alla presentazione del metodo effettivamente applicato dagli autori nella gestione di team di progetto. Mostra come, scegliendo opportunamente tra le tecniche indicate, sia possibile definire una metodologia di gestione dei progetti software basata su dati obiettivi e adatta alla propria organizzazione.

## Chi sono gli autori

**Domenico Clerici** - Laureato in Scienze dell'Informazione presso l'Università di Pisa nel 1979. Da allora si occupa dello sviluppo di applicativi bancari, oltre che di ingegneria del software, analisi metrica e qualità del software. Nel 2001 ha costituito TICLE Srl, di cui attualmente è amministratore.

**Luigi Petruzzelli** - Laureato con lode in matematica, si occupa di assicurazione di qualità e ingegneria del software dal 1994. Dal 1998 opera come libero professionista, svolgendo attività di consulenza principalmente sul controllo di progetti di sviluppo software e l'analisi metrica del software.

## Perché leggere *La gestione quantitativa di progetti software*?

- Per impadronirsi dei concetti fondamentali della gestione di progetti in ambito software
- Per essere in grado di predisporre all'interno della propria organizzazione un metodo quantitativo di previsione dell'andamento dei progetti (impegno e costi).
- Per essere in grado di selezionare metriche che "misurino" prodotti, progetti, processi software.
- Come introduzione al miglioramento dei processi aziendali per accrescerne l'efficienza, in particolare nell'ambito dello sviluppo di software.

## A chi si rivolge

- A chi desidera una visione generale delle più comuni tecniche di pianificazione e controllo di progetti in generale e di progetti software in particolare.
- A chi desidera vedere quali tecniche di project management siano effettivamente utilizzate all'interno di un'organizzazione che sviluppa software.
- A coloro che stanno implementando un corso di formazione sulla gestione di progetti software, come possibile ausilio didattico.
- Ai responsabili di gruppi di lavoro di unità produttive piccole o medie.
- A chi occupa un ruolo di capoprogetto o simile e desidera formalizzare le proprie conoscenze.
- Le figure di più alta responsabilità possono vedere com'è possibile ottenere previsioni di massima fin dalle fasi iniziali. Non è per loro tutto il libro, ma solo alcune parti selezionate.
- I progettisti / sviluppatori possono farsi un'idea di cosa succede "al contorno".

## Su cosa è basato

- Tecniche standard di project management riportate in letteratura.
- Tecniche statistiche di base, insieme a un po' di matematica.
- Letteratura e standard relativi a project management, gestione dei rischi, ingegneria del software, process improvement, misurazione di processi e prodotti software.
- Quasi quindici anni di esperienza diretta nell'affinamento e nella concreta applicazione in ambito produttivo di un metodo per la gestione dei progetti e la previsione dell'impegno.

## Conoscenze richieste

- Per il project management classico e la presentazione del modello principale, matematica e statistica di base (media, deviazione standard).
- Per alcune sezioni più avanzate, conoscenze anche non rigorose di statistica (in particolare, distribuzioni di probabilità).
- Per un approfondimento dei modelli proposti, nozioni basilari di calcolo differenziale e integrale.
- Per alcune esemplificazioni, conoscenza di base dei processi produttivi utilizzati nell'ambito dello sviluppo software.
- Per il miglioramento, visione generale (anche a livello elementare) dei processi presenti in ambito aziendale.
- In alcuni capitoli è preferibile, ma non indispensabile, avere almeno qualche mese di esperienza lavorativa.

## Dalla prefazione

Questo libro nasce da un'opera incompiuta. Nel 2006 abbiamo pubblicato il volume *Knowledge Based Project Management*, dedicato da un lato alla gestione di progetti generici, dall'altro alla presentazione di quanto poteva essere di interesse generale nell'ambito della nostra esperienza di gestione dei progetti software. Fin dalla sua pubblicazione avremmo voluto includere altre parti, appunto quelle rivolte più specificamente allo sviluppo di software. Ma il tempo era tiranno, e lo spazio limitato; così abbiamo dovuto aspettare un po' prima di riuscire a preparare il volume che avete in mano.

Riallacciandoci sia alla nostra esperienza, sia allo spirito di alcuni punti dello standard ISO 9001, che pone in risalto l'importanza della misurazione, abbiamo voluto presentare una panoramica della gestione quantitativa relativa allo sviluppo di software.

Nessuno dei capitoli è esaustivo, e tutti devono intendersi come un primo approccio agli argomenti proposti. Su ciascuno di questi potrebbe essere scritto almeno un libro, e in più di un caso lo è stato. Partiamo da una vista d'insieme sulla gestione dei progetti e sui cicli di vita del software, che ci permette sia di fornire una breve introduzione sia di precisare la terminologia impiegata nel seguito del libro. Si passa quindi ai punti tradizionali in cui i metodi quantitativi fanno la parte del leone: stime, rischi, modelli. Si prosegue con una presentazione dei processi aziendali e del loro miglioramento, nonché dei principali standard a essi collegati per quanto concerne il software.

Si descrivono poi le metriche: di processo e progetto, ma anche di prodotto. Quindi un piccolo esempio di quella che è la nostra esperienza nell'applicare metodi quantitativi alla gestione di piccoli gruppi di progetto (fino a otto persone) per una trentina di persone complessive.

Conclude il volume una succinta guida metodologica per scegliere gli strumenti software più adatti. Il libro, organizzato come descritto di seguito, si rivolge a chi desidera delle informazioni di base su come gestire quantitativamente processi o progetti di sviluppo software. I punti di maggior interesse potranno poi essere approfonditi tramite la vasta letteratura esistente.

## **Cap. 1 - Richiami di project management**

Per poter parlare di gestione quantitativa dei progetti occorre per prima cosa affrontare la gestione dei progetti. Ci proponiamo di introdurre soltanto i concetti basilari del project management per presentarne una visione introduttiva e di insieme, oltre che per definire una terminologia comune.

- Introduzione al project management
  - perché il project management?
  - il ciclo di vita
  - l'organizzazione
- Project management classico, estremo, adattativo
- La nascita di un progetto
  - gli obiettivi
  - criteri di successo del progetto
- L'arte della stima
  - la Work Breakdown Structure
  - tempi
  - risorse
  - costi
- Le persone e la composizione dei gruppi di lavoro
  - composizione dei gruppi di lavoro
  - le altre risorse
- I rischi
- La tempificazione
  - il reticolo del progetto
  - cenni a CPM e PERT
  - assegnazione e bilanciamento delle risorse
  - anticipare o posticipare la conclusione del progetto
  - GANTT
  - assegnare i lavori
  - probabilità di concludere in tempo
- Il controllo dell'avanzamento
  - il reporting
  - il controllo
  - stato di avanzamento lavori (SAL)
  - la Earned Value Analysis
  - azioni correttive e ripianificazione
- Le modifiche e la manutenzione
- La chiusura del progetto

## **Cap. 2 - Cicli di vita e approcci allo sviluppo di software**

Un progetto di sviluppo software non deve essere affrontato “a caso”, ma scegliendo con cognizione di causa l'approccio più adatto alle sue caratteristiche e alla propria organizzazione. Presentiamo una panoramica dei più diffusi cicli di vita, o anche più semplicemente approcci allo sviluppo, da quelli classici e consolidati alle tendenze attuali.

- Ciclo di vita a cascata
- Ciclo di vita a “V”
- Metodi incrementali / iterativi
  - ciclo di vita a spirale
  - prototipazione
  - RAD: Rapid Application Development
- Metodi “agili”
  - scrum
  - XP: Extreme Programming
  - FDD: Feature Driven Development
- i non-metodi

### **Cap. 3 - Metodi di stima**

Dopo aver incontrato i nostri attori, possiamo finalmente passare al primo esempio di metodi quantitativi: la stima di un progetto software (tempi, costi, risorse), in relazione alla quale forniremo una panoramica delle principali tecniche. L'esperienza delle persone deputate alla stima è insostituibile, ma può essere validamente affiancata da metodologie che si basano su dati obiettivi e modelli matematici.

- Metodi empirici e metodi analitici
- Cosa stimare?
- Stime per analogia
- Stime basate sul giudizio di esperti
- Stime top-down e bottom-up
- Punti funzione e loro evoluzione
  - i punti funzione di Albrecht
  - evoluzione dei punti funzione
- COCOMO
- Cenni ad altri metodi
  - il modello teorico di Norden
  - il modello di Putnam
- Stima e metodi "agili"
- I non-metodi

### **Cap. 4 - I rischi**

Non importa quanto una pianificazione sia stata accurata, un progetto è comunque sottoposto a rischi. Il problema è capire quanto i rischi siano elevati, cosa comporti il loro concretizzarsi e come si possano individuare le minacce prima che si attuino, intraprendendo le azioni più opportune per contrastarle. Un "rischio" particolare per le fasi successive al rilascio di un progetto sono i difetti, e nella parte conclusiva del capitolo presentiamo alcuni modelli utili per prevederli.

- La valutazione del rischio
  - liste di controllo
  - matrici di rischio
  - metodo Monte Carlo
- Il controllo del rischio
  - pianificazione della gestione
  - monitoraggio e risoluzione
- La difettosità: cenni ai Reliability Growth Models

### **Cap. 5 - Costruzione di modelli basati su dati storici**

Un modello è una rappresentazione di un fenomeno che ne riproduce le caratteristiche di interesse. Il suo obiettivo è, sulla base di parametri in ingresso, fornire stime attendibili di una o più grandezze. Per esempio, sulla base delle dimensioni stimate di un progetto software, determinarne il costo. Qui offriamo una panoramica sui modelli in generale, e mostriamo quindi in particolare come sia possibile costruire modelli basati sui dati storici dei propri progetti.

- La definizione dei modelli
- Modello per lo scostamento preventivo/consuntivo
- Modello per la percentuale di attività del ciclo di vita
- Modello per gli indici di straordinario/assenza
- Modello per la distribuzione dell'impegno
- Cenni ai modelli lineari

## **Cap. 6 - Costruzione di modelli basati su considerazioni teoriche**

Presentiamo un secondo approccio alla costruzione dei modelli. In assenza di dati storici, possiamo ricavare i modelli basandoci sulla formulazione matematica di considerazioni teoriche, e quindi verificare quanto i nostri progetti si comportano come teoricamente previsto. Solitamente sarà poi necessaria una fase di “adattamento” dei modelli alla realtà progettuale.

- Il modello di Norden e la distribuzione di Rayleigh
- Cenni alla costruzione di un modello basato sul ciclo di vita
  - distribuzioni base per i modelli
  - composizione di distribuzioni
  - costruzione di un modello per un ciclo di vita nuove tecnologie

## **Cap. 7 - Una visione per processi**

Alla classica visione di un'organizzazione per aree aziendali si è da tempo affiancata una visione trasversale “per processi”, attualmente sempre più diffusa anche grazie alle norme della serie ISO 9000 e ad altri standard internazionali. Presentiamo dapprima una succinta vista dei processi che entrano in gioco nello sviluppo di software; intervenire su questi processi significa guadagnare in efficacia o efficienza, con un conseguente ritorno economico. Nella parte finale del capitolo descriviamo un metodo di validità generale per il miglioramento dei processi.

- Cos'è un processo
- I processi aziendali
- Il processo di project management
- Interazioni tra macroaree aziendali
- Il miglioramento dei processi
  - definire gli obiettivi
  - individuare il punto di partenza
  - implementare le azioni di miglioramento
  - verificare il punto di arrivo
  - il miglioramento continuo

## **Cap. 8 - Principali standard per la qualità e il miglioramento del software**

Riportiamo una succinta guida ai principali standard per la qualità e il miglioramento nell'ambito dei processi di sviluppo software. La parte principale è svolta naturalmente dagli standard ISO, *International Organization for Standardization*, che di fatto sono quelli più utilizzati in ambito industriale.

- ISO/IEC 9003: la guida per il software di ISO 9001
- ISO/IEC 12207: i processi per il software
- ISO/IEC 15504 “SPICE”: assessment per i processi software
- CMMI<sup>®</sup>
- ami
- ISO/IEC 14764: la manutenzione del software

## **Cap. 9 - Le metriche e l'introduzione di un programma di misurazione**

Oltre a definire modelli o metodi che forniscano valori, numerici o meno, per poter parlare di gestione quantitativa è indispensabile misurare i propri processi, progetti, prodotti. Ciò fornirà la base da un lato per conoscere obiettivamente la propria situazione attuale, e dall'altro per dimostrare i miglioramenti. Qui presentiamo alcune generalità sulle metriche, oltre a qualche traccia utile per introdurre un programma di misurazione nella propria organizzazione.

- Le metriche e le loro caratteristiche
- Introduzione di un programma di misurazione
- Gli standard ISO/IEC 9126 e ISO/IEC 14598
- I moduli di valutazione

## **Cap. 10 - Metriche di processo e di progetto**

Descriviamo alcune metriche utili per misurare progetti specifici o, più in generale, efficacia ed efficienza dei processi di sviluppo. Dato il gran numero di processi che entrano in gioco nella realizzazione del software, ci limitiamo a quelli che riteniamo prioritari, sperando comunque di offrire una traccia al lettore interessato a misurare anche gli altri.

- Gli assessment
- Alcune metriche di processo
  - Conformità delle attività*
  - Copertura delle review*
  - Incidenza di ciascuna fase*
  - Produttività*
  - Efficacia nella rimozione dei difetti*
  - Costo di risoluzione dei problemi*
- alcune metriche per il processo di manutenzione
  - Fix backlog*
  - Indice di backlog management (BMI)*
  - Fix difettosi*
  - Percentuale di fix eccedenti i tempi di risposta*
  - Durata media di risoluzione dei problemi*
  - Altre metriche per la manutenzione*
- Alcune metriche di progetto
  - Deviazione dalle previsioni*
  - Tipo di progetto*
  - Impegno di sviluppo del progetto*
  - Funzioni da modificare / funzioni implementate*
  - Funzioni implementate / funzioni richieste*
  - Profilo della rimozione dei difetti*
  - Densità dei difetti durante il testing*
  - Profilo della segnalazione dei difetti durante il testing*

## **Cap. 11 - Metriche di prodotto**

Mentre si è generalmente concordi sull'impiego di metriche di progetto o di processo, l'utilizzo di metriche di prodotto è abbastanza controverso. Riteniamo tuttavia che l'uso di alcune metriche di prodotto selezionate possa contribuire al miglioramento della qualità del software. Perciò presentiamo le metriche più note, limitandoci a citarne altre più "esotiche" o meno diffuse.

- Metriche di base
  - LOC (linee di codice)*
  - Complessità ciclomatica di McCabe*
  - LOC commento / LOC*
  - Nomi autodescrittivi / nomi*
  - Salti incondizionati*
  - LOC / modulo*
  - Depth of Nesting*
  - Fan-out*
  - Fan-in*
- Altre metriche significative
  - Profondità architetturale*
  - Deviazione da un albero*
  - Maximum Depth of Nested Loops*
  - Relazione complessità interna / esterna*
  - Static Path Count*
  - Indice di copertura dei test*
  - La Software Science di Halstead*
  - Metriche di conteggio*
  - Alcune altre metriche*
- Metriche relative all'affidabilità
  - Densità dei difetti*
  - MTTF e MTBF*
  - Guasti / ore d'uso*
  - PUM*
- Cenni alle metriche per il software orientato agli oggetti
  - WMC*
  - DIT*
  - NOC*
  - RFC*
  - CBO*
  - LCOM*

## **Cap. 12 - Una metodologia di project management basata sui modelli**

Dopo aver trattato nelle linee essenziali la gestione quantitativa dei progetti software, indichiamo un possibile modo di monitorarli. L'intento del capitolo è mostrare come, scegliendo opportunamente tra le tecniche indicate, sia possibile individuare una metodologia di gestione dei progetti basata su dati obiettivi e adatta alla propria organizzazione.

- L'importanza della memoria
- L'organizzazione
- Il ciclo di vita
- La pianificazione
- Il reporting
- Il controllo e le azioni correttive
- Interazioni con gli altri processi
- Il miglioramento



## **Cap. 13 - Piccola guida alla scelta degli strumenti di supporto**

Per mettere in pratica quanto visto finora servono degli strumenti di supporto. Ne esistono di vari tipi: come scegliere il più adatto alle proprie esigenze? Forniamo una piccola guida per orientare la scelta; non abbiamo volutamente inserito un elenco ragionato degli strumenti presenti sul mercato, in quanto sono in continua evoluzione.

- Come selezionare uno strumento
- Determinare le proprie esigenze
- Altri aspetti da tenere in considerazione
- Costruzione di una griglia di valutazione
- Scelta dello strumento

## **Appendici – Glossario e bibliografia**

## Introduzione alla metodologia descritta nel Cap. 12

Nel capitolo dedicato alla metodologia gli autori hanno voluto indicare come hanno affrontato la gestione dei progetti da loro coordinati. Si tratta di una vista a livello abbastanza alto, come supervisione di più progetti contemporaneamente, affiancata a una visione più “capillare” per il singolo progetto. Questa metodologia consente una previsione “accurata ad alto livello”, rapida e basata su poche informazioni; inoltre è stato sviluppato un tool a suo supporto.

Riassumiamo qui molto succintamente parte del contenuto del capitolo dedicato alla metodologia.

- Il punto chiave è la memoria: in base all’andamento dei progetti già realizzati, si stimano i progetti futuri.
- I responsabili non devono comprendere gli aspetti teorici sottostanti; dovrebbero comunque comprendere che ogni metodologia ha un ambito di applicazione e dei limiti.
- Approccio metodologico orientato a
  - vista d’insieme del portafoglio progetti, inclusa la definizione di budget e risorse necessarie;
  - previsione di impegno, risorse e costi fin dalle prime fasi;
  - controllo degli scostamenti progettuali;
  - ottimizzazione dell’impegno delle persone.

I punti cardine della metodologia sono tre:

- l’esistenza di un ciclo di vita ben definito;
- un modello di riferimento per la distribuzione dell’impegno;
- un metodo di reporting del personale.

Comunque, la stima dell’impegno e dei tempi è lasciata ai responsabili tecnici (capi-progetto), così come il controllo a livello del gruppo di lavoro per approfondire le situazioni sospette evidenziate a livello più alto.

Passi principali:

- identificare un modello adeguato al progetto;
- in base all’impegno complessivo, alla data di inizio lavori e alla data di rilascio, stimare automaticamente la distribuzione dell’impegno per tipo di attività e la distribuzione mensile dell’impegno complessivo;
- utilizzare queste stime già in fase di offerta;
- a progetto avviato, reporting periodico del personale;
- produrre consuntivi “in corso d’opera” del progetto;
- confrontare i consuntivi con le previsioni;
- esaminare attentamente gli scostamenti;
- introdurre eventuali azioni correttive e ripianificare;
- aggiornare periodicamente i parametri del modello.

## Come ottenere *La gestione quantitativa di progetti software*?

Si può ordinare il volume al costo di 26 euro + spese di spedizione scrivendo a [ordini@edizionidellavigna.it](mailto:ordini@edizionidellavigna.it)

ISBN 978-88-6276-014-0; brossurato, formato 15x21; 260 pagine con illustrazioni in bianco e nero.