

Introduzione al Capability Maturity Model

*Linee guida per la comprensione del modello
SEI-CMM e la sua applicazione nelle aziende
informatiche di software di piccole e medie
dimensioni*

Ercole F. Colonese

Versione v1.0 - Settembre 2005

© SEI e CMM sono marchi registrati.

Note sulla versione attuale

La presente versione (V1.0) del documento descrive un approccio, semplice ed efficace, per l'applicazione nelle piccole e medie imprese del modello SEI/CMM per il miglioramento del processo di sviluppo del software.

L'approccio descritto si basa sull'esperienza maturata dall'autore in un progetto che ha visto il coinvolgimento di un gruppo di aziende informatiche italiane di piccole e medie dimensioni per il miglioramento dei processi produttivi e gestionali tramite l'introduzione delle "best practice" suggerite dal modello in oggetto (CMM) nella versione disponibile alla data di realizzazione del progetto (2001).

E' prevista una seconda edizione con gli aggiornamenti relativi alla versione attuale del modello (CMMI, v1.2, 2006)

2005 V1.0	2006	2007	2008	2009
--------------	------	------	------	------

Copyright © Ercole F. Colonese 2005

INDICE

Argomenti

Introduzione al Capability Maturity Model.....	5
1.1. Introduzione.....	7
1.1.1 Contenuti	7
1.1.2 Breve storia.....	7
1.1.3 Scopo del suo utilizzo.....	8
1.1.4 Concetti fondamentali della “Maturità dei processi”	9
1.2. Struttura del “Capability Maturity Model”	12
1.2.1 I cinque livelli di maturità	12
1.2.2 Key Process Areas (KPA).....	13
1.2.3 Evoluzione dei processi.....	17
1.3. Implementazione del “Capability Maturity Model”	19
1.3.1 Modalità implementative.....	19
1.4. Valutazione del livello di maturità CMM	22
1.4.1 Introduzione.....	22
1.4.2 Domande (Checklist).....	23
1.4.3 Rapporto di valutazione finale	30
Bibliografia.....	31

Figure

Figura 1. Elementi base della metodologia proposta.	9
Figura 2. Rappresentazione grafica dei livelli di maturità CMM.....	13
Figura 3. Struttura del modello CMM.....	20

Tabelle

Tabella 1. Descrizione dei livelli di maturità del modello CMM.	12
Tabella 2. Processi principali previsti nei livelli di maturità del modello CMM....	13
Tabella 3. Raggruppamento delle 18 KPA del modello CMM nelle tre categorie previste: gestione, organizzazione, ingegnerizzazione.....	18
Tabella 4. Modalità implementative del modello CMM.....	19
Tabella 5. Lista di controllo (checklist) per la valutazione del livello di maturità CMM.	23
Tabella 6. Rapporto di valutazione finale.	30

Introduzione al Capability Maturity Model

Linee guida per la comprensione del modello e la sua applicazione nelle PMI informatiche di software

1.1. Introduzione

1.1.1 Contenuti

Il presente documento fornisce gli elementi di base del *Capability Maturity Model* definito dal *Software Engineering Institute* (nel seguito nominato più semplicemente come SEI/CMM), considerato il modello di riferimento nel mondo dello sviluppo del software negli USA, e alcuni suggerimenti pratici, concreti ed efficaci per la sua applicazione alle piccole e medie imprese che sviluppino software.

Quanto descritto si basa sull'esperienza maturata dall'autore in un progetto italiano di grande portata che ha visto impegnato un gruppo di circa 30 piccole e medie imprese (PMI) informatiche nell'accrescere il proprio livello di competitività adottando proprio tale modello per migliorare i propri processi produttivi e gestionali nel campo dello sviluppo software.

I risultati positivi conseguiti dalle aziende coinvolte permettono di suggerire ad altre aziende simili (aziende di software di piccole e medie dimensioni) quanto sperimentato e proposto in questo documento.

1.1.2 Breve storia

Il *Capability Maturity Model* (CMM) è stato ideato ed introdotto inizialmente da Watts Humphrey, ai tempi in cui lavorava in IBM per il Ministero della Difesa, con l'aiuto di un gruppo di professionisti del software. Su esplicita richiesta del Ministero, il modello è stato poi acquisito, sviluppato e sponsorizzato dalla *Carnegie Mellon University – Software Engineering Institute* (SEI).

Il modello di sviluppo SEI/CMM parte dall'assunzione che la qualità del software dipende fortemente dal processo utilizzato per il suo sviluppo e per la successiva manutenzione.

In estrema sintesi, il modello è l'applicazione con buon senso delle migliori tecniche di gestione dei processi e del miglioramento della qualità. Si basa principalmente su:

- linee guida comuni per lo sviluppo e la manutenzione del software (condivise da centinaia di produttori e da enti governativi);
- struttura per la valutazione consistente dei livelli raggiunti.

Il "processo software" è definito dal SEI/CMM come un insieme di attività, metodi, pratiche e trasformazioni che le aziende di software utilizzano per sviluppare e per mantenere il software ed i prodotti associati. Nel seguito del documento il processo software è più esplicitamente nominato come "processo di sviluppo del software".

Nota: Il modello SEI/CMM non indirizza tutti i processi relativi al software e le tematiche relative al miglioramento della qualità. Esso non indirizza, per esempio, oppure indirizza solo indirettamente, le tematiche relative a:

- specifici tool, metodi e tecnologie (es. Object Orientation, Java, ecc.);
- ingegnerizzazione dei sistemi, pratiche di marketing;
- risorse umane, comportamenti organizzativi, ecc.

Esso è un modello per studiare, valutare e migliorare i processi per lo sviluppo del software.

Il modello SEI/CMM sta acquistando sempre più valore in Europa e nel nostro paese negli ultimi tempi. Prima totalmente sconosciuto, ora il modello è sempre più discusso sulla stampa specializzata ed è oggetto di sperimentazione da parte di aziende informatiche leader nel mercato nazionale ed estero.

1.1.3 Scopo del suo utilizzo

Come detto in precedenza, il modello SEI/CMM è utilizzato principalmente per valutare un processo di sviluppo del software e per pianificarne le modifiche migliorative. Esso può essere utilizzato senza alcun coinvolgimento del SEI. Il coinvolgimento diretto di valutatori certificati è necessario solo per ottenere la certificazione¹ del livello di maturità raggiunto.

Scopo principale dell'adozione del modello SEI/CMM è quello di migliorare i processi di sviluppo del software in ottica di:

- miglioramento della qualità del software prodotto;
- aumento della produttività dell'organizzazione di sviluppo;
- riduzione dei tempi di sviluppo (riduzione del ciclo di vita del software).

Il progetto qualità svolto in Sicilia ha utilizzato con successo il modello SEI/CMM come base di riferimento per proporre le modifiche migliorative ai processi esistenti presso le aziende coinvolte e per valutarne i risultati conseguiti, misurando il livello di maturità raggiunto da ciascuna azienda relativamente ai singoli processi considerati. Per singoli processi si intendono in questo contesto i sottoprocessi del ciclo di sviluppo del software come:

- Analisi e Disegno;
- Codifica e Validazione (Testing);

¹ In realtà non si tratta di una vera e propria certificazione (come quelle rilasciate, per esempio, per l'ISO9000 da parte degli Enti certificatori preposti), ma di una valutazione fornita dai "Assessor" riconosciuti dal SEI relativamente al livello di maturità raggiunta dall'organizzazione rispetto al modello preso come riferimento.

- Rilascio e Manutenzione;
- Gestione dei progetti (Project Management);
- Gestione della configurazione e delle modifiche (Change and Configuration Management);
- Controllo della qualità (Quality Assurance);
- Introduzione delle nuove tecnologie (Technology Change Management).

L'attività consulenziale e quella di valutazione finale è stata svolta da consulenti qualificati e non ha previsto il rilascio di alcuna certificazione.

1.1.4 Concetti fondamentali della “Maturità dei processi”

I tre elementi principali della maturità dei processi (vedi Figura 1 di seguito) sono:

1. Persone con le competenze richieste, la formazione necessaria e la giusta motivazione;
2. Metodi e procedure con relazione alle singole attività da svolgere (task);
3. Strumenti ed apparecchiature.

Un processo efficace per lo sviluppo del software deve integrare questi tre elementi in maniera armonica ed adeguati alla struttura aziendale.

Di seguito è descritto come i tre elementi sono stati interpretati ed applicati nel progetto qualità in Sicilia nel caso di piccole e medie imprese informatiche.

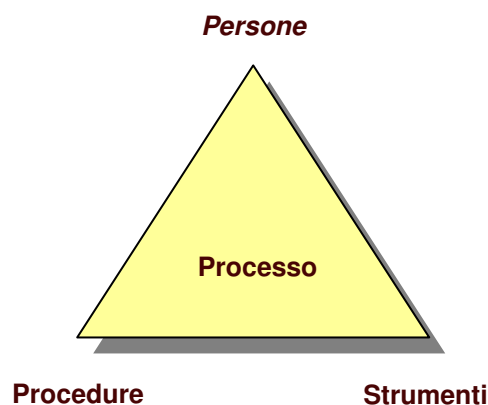


Figura 1. Elementi base della metodologia proposta.

Persone

Il coinvolgimento di persone con le competenze richieste nel processo di sviluppo del software è fondamentale per la riuscita dei progetti. Accanto alle competenze richieste è necessario definire anche i ruoli e le responsabilità di ognuno. Ciò per assicurare che ciascuno sappia cosa fare e conosca le proprie responsabilità. A questo scopo sono state definiti i ruoli principali coperti con le relative responsabilità assegnate e le competenze richieste. I ruoli previsti sono in numero inferiore a quanto generalmente si può trovare nelle grandi aziende. Alcuni ruoli, in particolare, sono coperti dalle stesse persone (per esempio il capo progetto è anche l'analista esperto responsabile del disegno; l'assicurazione della qualità è garantita dal responsabile dell'azienda; l'analista-programmatore è anche il testatore; ecc.).

Procedure

Un processo ben definito è il secondo elemento qualificante di una metodologia efficace per lo sviluppo del software. Esso richiede che siano descritti le attività da svolgere, le modalità e le tecniche da adottare, gli output da produrre. Il ciclo di sviluppo del software che sia snello ed efficace non può che essere quello iterativo incrementale. Quello sperimentato in Sicilia prevede le quattro fasi fondamentali: 1) Analisi e disegno, 2) Programmazione e test unitario, 3) Test di integrazione e collaudo, 4) Rilascio e Manutenzione post rilascio. Parallelamente alle quattro fasi precedenti è prevista la fase di Gestione del progetto (Project Management) che include la gestione dei fornitori ed il controllo della qualità. Ogni fase è descritta in termini di: a) descrizione sintetica, b) input richiesti, c) attività da svolgere, d) output da produrre, e) validazione da effettuare, f) ruoli coinvolti, g) metodi, tecniche e strumenti da adottare, h) registrazioni da conservare per la dimostrazione della conformità (registrazione della qualità secondo le norme ISO9000).

Strumenti

La disponibilità di strumenti adeguati a supporto dello sviluppo del software costituisce il terzo elemento di base del modello SEI/CMM proposto. Essi, infatti, consentono di agire sulla produttività e sulla qualità dello sviluppo del software. A questo scopo, il processo proposto e sperimentato dalle aziende siciliane prevedeva l'adozione di apposite "strumenti" (chiamati "Tecniche") costituiti da metodi, tecniche, modelli, tool, metriche, ecc. da utilizzare durante le diverse fasi del ciclo di sviluppo del software.

In aziende software di piccole e medie dimensioni, dove il bilancio non permette errori di investimento, è di estrema importanza (vitale) un corretto equilibrio fra le tre componenti. Anche una sola persona può fare la differenza, l'utilizzo di uno strumento adatto può aumentare la produttività e l'adozione di un processo di sviluppo snello ma efficace può ridurre notevolmente i tempi di rilascio del software prodotto.

In altre parole si può riassumere il tutto con il motto “fare bene al primo colpo!”. Nel mercato sempre più competitivo e con esigenze crescenti ci sono ristretti margini di manovra, non si ha tempo e risorse per ripetere quanto fatto male nel ciclo precedente.

1.2. Struttura del “Capability Maturity Model”

1.2.1 I cinque livelli di maturità

Il modello SEI/CMM definisce cinque livelli di maturità crescente del processo di sviluppo del software. La definizione dei singoli livelli è riportata nella tabella che segue.

Tabella 1. Descrizione dei livelli di maturità del modello CMM.

Livello	Descrizione
1. Iniziale	Il processo di sviluppo software è caratterizzato da una scarsa strutturazione, spesso è assente, a volte è caotico. Solo pochi processi sono definiti ed il successo dipende dall’impegno dei singoli, a volte “eroico”.
2. Ripetibile (Gestito)	Sono stabili processi base di gestione dei progetti di sviluppo software per tracciare i costi, la schedulazione delle attività e le funzionalità sviluppate. Il processo è stabilito per essere ripetibile su progetti che sviluppano applicazioni simili.
3. Definito	Il processo di sviluppo software, sia per la parte di gestione che per quella di sviluppo tecnico, è definito, documentato, standardizzato ed integrato in azienda per l’intera organizzazione. Tutti i progetti utilizzano una versione del processo approvata, standardizzata ed adattabile ai singoli progetti.
4. Gestito quantitativamente	Si effettuano misure sul processo di sviluppo software e sulla qualità dei prodotti sviluppati. Sia il processo di sviluppo che i prodotti sviluppati sono quantitativamente interpretati e controllati.
5. Ottimizzato	Esiste un processo di miglioramento continuo basato su feedback quantitativi provenienti dall’utilizzo del processo e dalla sperimentazione di progetti pilota su innovazioni e nuove tecnologie.

Il modello SEI/CMM presuppone che le aziende crescano progressivamente sulla scala di maturità man mano che consolidano i vari livelli previsti adottando efficacemente i singoli processi previsti per ciascun livello.

Un risultato di sicuro valore competitivo è quello di raggiungere il livello 3 di maturità dove i processi principali sono definiti ed adottati efficacemente in tutti i progetti. A livello 3 le aziende sono in grado di ottenere i risultati attesi.

Il modello SEI/CMM indirizza in parte anche i requisiti delle norme UNI EN ISO 9001:1994 e fornisce, in aggiunta, alcuni elementi qualitativi tipici dello sviluppo del software. Per questo motivo il livello 3 è spesso associato al livello delle

aziende certificate ISO9000. Ciò è errato in termini specifici ma è un buon elemento di equivalenza. In particolare, dall'analisi di un'indagine condotta qualche anno fa, risulta che solo una piccola percentuale di aziende informatiche nel mondo raggiunge il livello 5 ed il livello 4, mentre la stragrande maggioranza delle aziende si colloca nei livelli 1 e 2. Il livello 3, associato, come già detto, alla certificazione ISO9000, raggruppa quindi un numero sempre crescente di aziende software e rappresenta un elemento importante di competitività.

La figura che segue mostra la tipica rappresentazione grafica dei livelli di maturità come gradini di una scala da salire verso l'eccellenza.

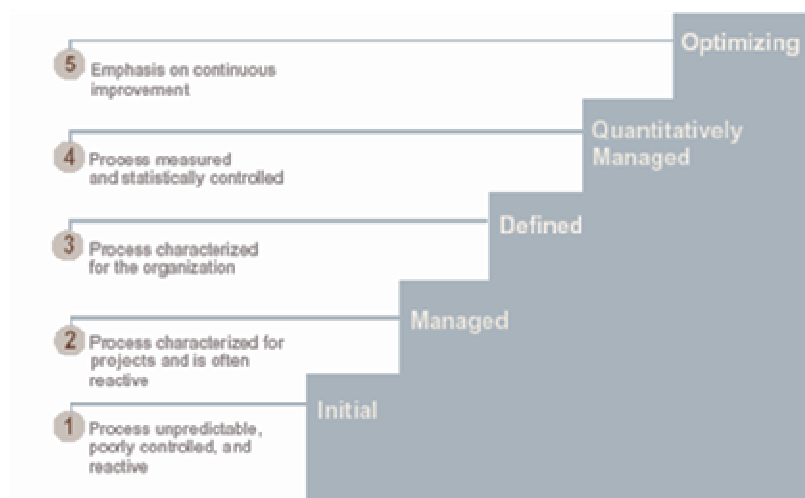


Figura 2. Rappresentazione grafica dei livelli di maturità CMM.

1.2.2 Key Process Areas (KPA)

Descrivono “cosa” fare e rappresentano i processi basilari dello sviluppo del software. Il modello SEI/CMM identifica 18 processi basilari (KPA) e li associa ai diversi livelli di maturità come mostrato nella tabella che segue.

Tabella 2. Processi principali previsti nei livelli di maturità del modello CMM.

Livello	Key Process Area (KPA)
1. Iniziale	Nessuno processo
2. Ripetibile (Gestito)	1. Gestione dei requisiti 2. Pianificazione dei progetti software 3. Controllo dei progetti software 4. Gestione dei fornitori coinvolti nei progetti software 5. Controllo della qualità del software (Quality Assurance) 6. Gestione della configurazione del software

Livello	Key Process Area (KPA)
3. Definito	7. Attenzione al processo organizzativo 8. Definizione del processo organizzativo 9. Programma di addestramento 10. Gestione dell'integrazione del software 11. Ingegnerizzazione del prodotto software 12. Coordinamento dei gruppi di lavoro 13. Revisioni paritetiche (Peer reviews)
4. Gestito quantitativamente	14. Gestione degli aspetti quantitativi dei processi 15. Gestione della qualità del software
5. Ottimizzato	16. Prevenzione dei difetti 17. Gestione delle modifiche tecnologiche 18. Gestione delle modifiche ai processi

Segue una breve descrizione dello scopo principale dei processi chiave previsti nei diversi livelli. Per maggiori dettagli sui singoli processi è bene consultare il modello nella sua versione originale².

Key Process Area del livello 2

Scopo principale del processo di *Gestione dei requisiti* è quello di stabilire una comune base di intesa tra il cliente e l'azienda di software sui requisiti da sviluppare. Tale accordo sui requisiti deve essere la base per la pianificazione e la gestione del progetto di sviluppo software.

Scopo principale del processo *Pianificazione dei progetti* è quello di stabilire piani di sviluppo realistici in accordo con i requisiti concordati con il cliente e con le capacità di sviluppo dell'azienda di software.

Scopo principale del processo di *Controllo dei progetti* è quello di stabilire una adeguata visibilità dello stato di avanzamento del progetto in modo da permettere al management di intraprendere le adeguate azioni correttive in caso di deviazione dai piani di sviluppo stabiliti.

Scopo principale del processo *Gestione dei fornitori di software* è quello di selezionare terze parti qualificate e di gestirle in maniera efficace.

Scopo principale del processo *Assicurazione della qualità del software* è quello di fornire al management la giusta visibilità del processo di sviluppo adottato per lo sviluppo del software e la qualità dei prodotti intermedi (output) realizzati.

² Purtroppo non esiste alla data una versione in lingua italiana del modello.

Scopo principale del processo *Gestione della configurazione del software* è quello di stabilire e di mantenere l'integrità del prodotto software sviluppato durante l'intero ciclo di vita del prodotto (ciclo di sviluppo).

Key Process Area del livello 3

Scopo principale del processo *Attenzione al processo organizzativo* è quello di stabilire le responsabilità all'interno dell'organizzazione relativamente alle attività del processo di sviluppo del software in modo da migliorarne le capacità produttive nel suo insieme.

Scopo principale del processo *Definizione del processo organizzativo* è quello di sviluppare e mantenere un insieme di *asset* aziendali relativi al processo di sviluppo del software (regole, metodi, strumenti, ecc.) in grado di migliorare le capacità produttive dell'azienda su tutti i progetti software. Tali asset costituiscono la base per una gestione quantitativa del processo di sviluppo del software in ottica di miglioramento continuo. Gli stessi asset costituiscono un patrimonio importante dell'azienda e sono divulgati tramite piani di formazione opportuni.

Scopo principale del processo *Programma di addestramento* è quello di sviluppare le competenze e le conoscenze delle persone coinvolte nel processo di sviluppo del software in modo che esse possano svolgere con efficacia il ruolo loro assegnato. Tale programma è di responsabilità della direzione aziendale mentre ogni progetto ha la responsabilità di definire le competenze richieste di e di formarne quelle eventualmente specifiche (uniche) del progetto in questione.

Scopo principale del processo *Gestione dell'integrazione del software* è quello di integrare le attività di ingegnerizzazione del software (analisi, disegno e sviluppo) e le attività di gestione del progetto (project management) in un processo unico, definito, coerente, adattabile a tutti i progetti e basato sugli asset aziendali. L'eventuale personalizzazione per i singoli progetti è basata sullo specifico business da indirizzare e sugli aspetti tecnici specifici.

Scopo principale del processo *Ingegnierizzazione del prodotto software* è quello di eseguire in maniera consistente e ben definita per il singolo progetto la realizzazione del prodotto coerentemente con il processo generale di sviluppo del software stabilito. Ciò si opera attraverso le regole di personalizzazione stabilite per i singoli progetti tramite le quali sono definite le attività specifiche da eseguire (analisi dei requisiti, disegno, codifica, test, ecc.).

Scopo principale del processo *Coordinamento dei gruppi di lavoro* è quello di stabilire uno strumento che permetta al gruppo di sviluppo di cooperare efficacemente con altri gruppi di sviluppo in modo che il progetto possa meglio indirizzare i requisiti del cliente, più efficacemente e più efficientemente, per esempio indirizzando i requisiti di sistema, gli obiettivi e le criticità.

Scopo principale dell'area *Revisioni paritetiche* è quello di rimuovere gli errori ed i difetti del prodotto software prima possibile ed in maniera efficace.

Key Process Area del livello 4

Scopo principale del processo *Gestione degli aspetti quantitativi dei processi* è quello di controllare le performance del processo di sviluppo del software in maniera quantitativa. Ciò è fatto misurando i risultati conseguiti a fronte di quanto stabilito e quindi atteso.

Scopo principale del processo *Gestione della qualità del software* è quello di stabilire per ogni progetto una definizione quantitativa della qualità attesa del software e di controllare il raggiungimento di specifici obiettivi (quality goals) da parte del prodotto sviluppato. Ciò in base ai requisiti del cliente, degli utenti e dell'azienda.

Key Process Area del livello 5

Scopo principale del processo *Prevenzione dei difetti* è quello di identificare l'origine dei difetti, di analizzarne le cause e di mettere in opera opportune azioni che evitino in futuro (in altri progetti) il loro ripresentarsi. Spesso ciò comporta la modifica (il miglioramento) del processo di sviluppo del software.

Scopo principale del processo *Gestione delle modifiche tecnologiche* è quello di identificare i benefici delle nuove tecnologie disponibili e di introdurle nell'organizzazione attuale in maniera ordinata ed efficace.

Scopo principale del processo *Gestione delle modifiche ai processi* è quello di migliorare continuamente il processo di sviluppo del software con lo scopo di migliorare la qualità del software prodotto, di aumentare la produttività dell'organizzazione e di diminuire il tempo di realizzazione (ciclo di sviluppo).

Dei 18 sottoprocessi previsti dal modello SEI/CMM il progetto cui si fa riferimento in questo documento prevedeva l'implementazione della maggior parte di essi. Alcuni sono stati indirizzati da processi specifici (es.: Analisi dei requisiti, Project Management, Change and Configuration Management, Analisi-Disegno-Codifica, Revisione-Test-Collauda, ecc.). Altri, invece, sono indirizzati dalla progettazione di specifici centri di competenza permanente (es.: Centro di competenza sulle nuove tecnologie, ecc.).

1.2.3 Evoluzione dei processi

I 18 sottoprocessi (KPA) possono essere raggruppati in tre categorie distinte relativamente a:

- Gestione
- Organizzazione
- Ingegnerizzazione.

La tabella che segue mostra come i 18 KPA sono raggruppati nelle tre categorie di cui sopra a seconda dei cinque livelli crescenti di maturità (origine: documentazione ufficiale del SEI/CMM riportata in Bibliografia).

Manuali di sviluppo di software

Tabella 3. Raggruppamento delle 18 KPA del modello CMM nelle tre categorie previste: gestione, organizzazione, ingegnerizzazione.

Livello di maturità dei processi	Categoria dei processi		
	Gestione	Organizzazione	Ingegnerizzazione
5. Ottimizzato		<ul style="list-style-type: none"> Gestione delle modifiche tecnologiche 	
	<ul style="list-style-type: none"> Gestione delle modifiche ai processi 		<ul style="list-style-type: none"> Prevenzione dei difetti
4. Gestito	<ul style="list-style-type: none"> Gestione quantitativa dei processi 		<ul style="list-style-type: none"> Gestione della qualità del software
3. Definito	<ul style="list-style-type: none"> Gestione dell'integrazione del software Coordinamento dei gruppi 	<ul style="list-style-type: none"> Attenzione al processo organizzativo Definizione del processo organizzativo Programma di formazione 	<ul style="list-style-type: none"> Ingegnerizzazione dei prodotti software Revisioni paritetiche (Peer reviews)
2. Ripetibile	<ul style="list-style-type: none"> Gestione dei requisiti Pianificazione dei progetti Controllo dei progetti Gestione dei fornitori Assicurazione della qualità del software (QA) Gestione della configurazione 		
1. Iniziale	Processi ad hoc		

1.3. Implementazione del “Capability Maturity Model”

1.3.1 Modalità implementative

L’implementazione del modello SEI/CMM riguarda il “come” attuare il “cosa” fare, cioè come sviluppare i processi fondamentali (KPA) previsti per i vari livelli.

A tale scopo per ciascun sottoprocesso (KPA) sono definiti obiettivi, impegno ad eseguire, capacità di eseguire, attività da svolgere, misure ed analisi da effettuare, verifiche da eseguire sull’applicazione di quanto svolto dall’organizzazione.

Tabella 4. Modalità implementative del modello CMM.

Obiettivi	Descrivono gli obiettivi specifici che si vogliono raggiungere con l’adozione di una determinata pratica (KPA).
Impegno ad eseguire	Descrive le azioni che l’organizzazione deve compiere per assicurare che il processo di sviluppo del software sia stabilito e sia seguito da tutti. Tale responsabilità è della direzione aziendale ed è esplicitata nella “Politica per la qualità” scritta e distribuita a tutto il personale aziendale.
Capacità di eseguire	Descrive le condizioni che occorre assicurare perché il processo di sviluppo del software possa essere eseguito con efficacia. Generalmente si riferisce alle risorse (competenze richieste per i vari ruoli definiti), alla struttura organizzativa (persone e mezzi rese disponibili), alla formazione.
Attività svolte	Descrive le attività da svolgere, i ruoli coinvolti e le procedure da seguire per ciascun sottoprocesso.
Misure ed analisi	Descrive le misurazioni da eseguire per stabilire l’esatto stato di implementazione del processo di sviluppo del software. Tali misurazioni sono eseguite per controllare e migliorare il processo.
Verifica dell’applicazione	Descrive i controlli da eseguire per assicurare che le attività siano svolte in accordo con il processo di sviluppo del software stabilito. Generalmente si tratta di revisioni e controlli (audit) che la direzione aziendale delega ufficialmente al gruppo di Assicurazione Qualità (In termini ISO9000 potrebbe essere il Rappresentante dell’Assicurazione della Qualità – RAQ) che informa periodicamente la direzione stessa dei risultati.

La figura che segue mostra una rappresentazione grafica della struttura del modello CMM.

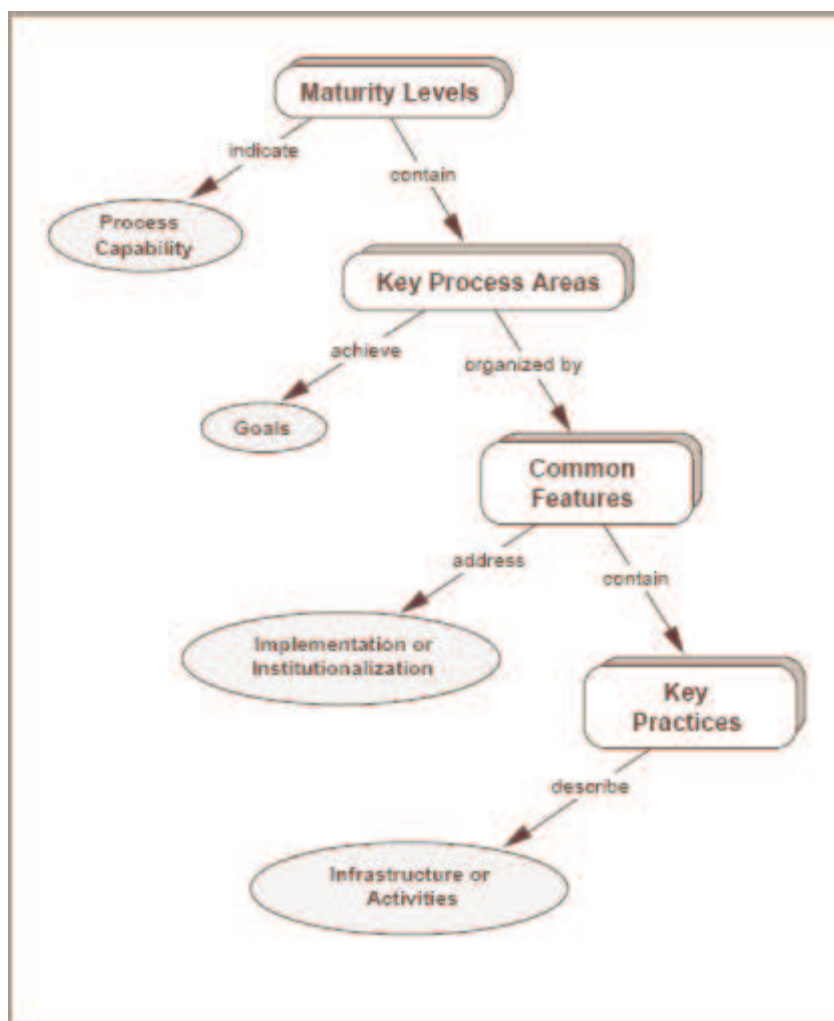


Figura 3. Struttura del modello CMM.

Ciascun “livello di maturità” (Maturity Level) indica la “capacità” che deve possedere ciascun processo “contenuto” nel livello (Process Capability).

Ciascuna “area di processo” (Key Process Area), a sua volta, ha obiettivi specifici (Goals) ed è organizzata in termini di “elementi” (“Common Features”). Questi elementi “contengono” delle pratiche (Key Practices) che è bene tener presente nello svolgimento del lavoro quotidiano in quanto costituiscono, a tutti gli effetti, delle “best practice” consolidate sul mercato e di sicuro successo.

Il modello CMM può essere semplicemente “implementato” in azienda oppure essere “istituzionalizzato”. In questo secondo caso, il modello mostra tutta la sua efficacia a livello dell’intera organizzazione.

Un processo può essere implementato con successo in un determinato modo in un progetto, ed in maniera diversa in un altro progetto. Entrambi i progetti possono

avere successo e dimostrare la validità dell'implementazione. Un processo è invece istituzionalizzato quando l'organizzazione decide di utilizzarlo lo stesso processo "tutti i progetti" stabilendo regole precise per la loro personalizzazione nei diversi contesti. Questo modo di operare garantisce l'uniformità operativa dell'intera organizzazione e permette di misurare l'efficacia del processo con metriche e tecniche statistiche, e di migliorare il processo stesso con opportune azioni.

Per una efficace implementazione del modello il progetto svolto in Sicilia ha definito, per ciascun processo (KPA) sperimentato, come adottare concretamente tali elementi. Ciò è stato documentato in termini di "Linee guide operative per l'implementazione in azienda delle metodologie definite". Ciascuna di esse propone alla direzione aziendale modalità specifiche ben definite per indirizzare efficacemente gli elementi sopraelencati (obiettivi, impegno ad eseguire, capacità di eseguire, attività svolte, misure ed analisi, verifica dell'applicazione) con esempi e personalizzazioni per le PMI. Essi sono descritti in un capitolo successivo.

1.4. Valutazione del livello di maturità CMM

1.4.1 Introduzione

Il *Software Engineering Institute (SEI)* stabilisce le modalità per la valutazione del livello di maturità raggiunto da un'organizzazione software relativamente ai singoli processi. In particolare, è previsto che la valutazione sia effettuata da personale specializzato, con esperienza pratica nelle tecniche di “assessment” e che sia qualificato dal SEI a svolgere tale attività valutativa. Solo i valutatori accreditati possono rilasciare una certificazione valida del livello di maturità raggiunto dall'azienda visitata.

Nel caso del progetto siciliano l'attività di valutazione finale è stata svolta dai consulenti e non è stato previsto il rilascio di alcuna certificazione.

Di seguito è riportata una lista di domande (checklist) che permettono di verificare il livello di maturità raggiunto da un'azienda che abbia adottato il modello SEI/CMM per migliorare il proprio processo di sviluppo del software. E' uno strumento utile sia per le singole aziende che vogliano effettuare un'autovalutazione del proprio livello di maturità raggiunto, sia da parte dei valutatori esterni che siano richiesti di effettuare una valutazione più soggettiva.

Essa non è un documento ufficiale del SEI ma uno strumento utile ad uso e consumo delle aziende e dei consulenti chiamati ad aiutarli nel processo di miglioramento continuo.

La lista è stata costruita sulla base dell'edizione del *Capability Maturity Model (CMM)* del 1991³. Le domande sono raggruppate in 5 sezioni diverse:

1. Organizzazione
2. Documentazione di standard e procedure
3. Metriche relative ai processi
4. Analisi e gestione dei dati
5. Controllo del processo.

Ogni sezione presenta domande specifiche ai diversi livelli di maturità (da 2 a 5). Alcune domande si riferiscono ad aspetti specifici assolutamente “richiesti” dalla

³ Alla data in cui viene pubblicato il presente documento esistono sicuramente altre liste di controllo (checklist) più aggiornate ed in linea con la nuova versione del modello. Qui si riporta la lista utilizzata nel progetto sperimentato per mostrare su quali elementi è stata condotta la valutazione. Nonostante la lista sia “datata” i suoi elementi di base rimangono tuttora validissimi e possono essere utilizzati dalle organizzazioni attuali per valutare (internamente) il proprio livello di “maturità” inteso come “capacità di “svolgere il lavoro in maniera produttiva ed efficace”.

metodologia. La risposta (Si o No) indica se l'argomento trattato dalla domanda è implementato in azienda oppure no.

Il livello di maturità è calcolato in base alla percentuale di risposte positive. La percentuale richiesta per raggiungere la maturità di ciascun livello è 80% di risposte generali positive e 90% di risposte positive richieste.

Prima di valutare un livello specifico (es.: livello 3) occorre valutare il livello precedente (es.: livello2).

In fondo sono fornite domande aggiuntive per la raccolta di informazioni di dettaglio.

A seguire è riportato il modulo da utilizzare come rapporto finale della valutazione effettuata dai consulenti. Esso permette di riassumere il livello di attuazione dei diversi principi del modello CMM:

- Obiettivi
- Impegno ad eseguire
- Capacità di eseguire
- Attività svolte
- Misure ed analisi
- Verifica dell'applicazione

Esso può essere opportunamente personalizzato per i singoli sottoprocessi (KPA) per meglio focalizzare le loro specificità, pur mantenendo l'approccio generale qui impostato. Così è stato fatto per ciascun processo indirizzato dal progetto qualità in Sicilia.

1.4.2 Domande (Checklist)

Tabella 5. Lista di controllo (checklist) per la valutazione del livello di maturità CMM.

Num.	Domanda	Liv.	Si/ No	Rich.
1.1	Organizzazione			
1.1.1	E' assegnato un responsabile per ciascun progetto di sviluppo software?	2		
1.1.2	Il responsabile di progetto riferisce direttamente alla direzione?	2		
1.1.3	Esiste una funzione di <i>Assicurazione Qualità</i> (QA) indipendente dal responsabile di progetto?	2		R
1.1.4	Esiste un responsabile competente per lo sviluppo delle Interfacce Utente?	3		

Manuali di sviluppo di software

Num.	Domanda	Liv.	Si/ No	Rich.
1.1.5	E' presente la funzione di <i>Ingegneria del software</i> nel gruppo di disegno?	3		
1.1.6	E' svolta da qualcuno la funzione di <i>Controllo della configurazione</i> ?	2		R
1.1.7	E' svolta da qualcuno la funzione di <i>Ingegneria del software</i> ?	3		R
1.2	Risorse, Personale, Formazione			
1.2.1	Ogni sviluppatore è dotato di una propria stazione di lavoro (PC)?	3		
1.2.2	Esiste un piano di formazione sulla gestione dei progetti software (<i>project management</i>) per ogni capo progetto?	2		
1.2.3	Esiste un piano di formazione sulle metodologie di sviluppo del software per ogni sviluppatore?	3		R
1.2.4	Esiste un piano di formazione sulle metodologie di sviluppo del software per ogni responsabili di progetto?	3		
1.2.5	Esiste un piano di formazione sulle metodologie di revisione ed ispezioni tecniche per ogni esperto che esegua tali attività?	3		R
1.3	Gestione della tecnologia			
1.3.1	E' usato un qualche meccanismo per mantenersi aggiornati sullo stato dell'arte delle tecnologie per lo sviluppo del software?	2		
1.3.2	E' usato un qualche meccanismo per valutare le tecnologie utilizzate in azienda rispetto a quelle disponibili sul mercato?	3		
1.3.3	E' usato un qualche meccanismo per decidere quando inserire in azienda una nuova tecnologia per lo sviluppo del software?	4		
1.3.4	E' usato un qualche meccanismo per la gestione ed il supporto durante l'introduzione in azienda di una nuova tecnologia?	4		R
1.3.5	E' usato un qualche meccanismo per identificare le tecnologie obsolete usate in azienda e sostituirle?	5		R
2.1	Documentazione di standard e procedure			
2.1.1	E' utilizzato un processo standardizzato e documentato per lo sviluppo del software in ogni progetto?	3		R
2.1.2	La documentazione del processo di sviluppo software standardizzato descrive l'utilizzo di tecniche e strumenti?	3		R
2.1.3	Esiste una procedura formale per la revisione manageriale di ciascun progetto di sviluppo software prima di firmare il contratto?	2		R
2.1.4	Esiste una procedura formale per la revisione periodica dello stato di ciascun progetto di sviluppo software?	2		
2.1.5	Esiste un meccanismo che assicuri che i fornitori coinvolti nello sviluppo di parti del progetto seguano un processo controllato per lo sviluppo del software?	2		
2.1.6	Sono utilizzati standard per i contenuti dei documenti relativi al software?	3		

Manuali di sviluppo di software

Num.	Domanda	Liv.	Si/ No	Rich.
2.1.7	Sono condotte revisioni indipendenti per ciascuna fase del processo di sviluppo?	2		
2.1.8	E' utilizzato qualche meccanismo per valutare se esiste un disegno o un codice riusabile nei nuovi progetti?	3		
2.1.9	Sono utilizzati standard di programmazione in ciascun progetto di sviluppo software?	2		
2.1.10	Sono utilizzati standard per la preparazione dei casi di test?	3		
2.1.11	Sono utilizzati standard per la manutenibilità del codice?	3		
2.1.12	Sono utilizzati standard durante la revisione tecnica del disegno?	4		
2.1.13	Sono utilizzati standard durante la revisione del codice?	4		R
2.1.14	E' utilizzata una procedura formale per la stima delle dimensioni del codice?	2		R
2.1.15	E' utilizzata una procedura formale per produrre la pianificazione dello sviluppo software?	2		R
2.1.16	E' utilizzata una procedura formale per valutare i costi dello sviluppo del software?	2		R
2.1.17	E' utilizzato un meccanismo per assicurare che i progettisti indirizzino i requisiti software?	2		
2.1.18	Sono applicati standard per lo sviluppo delle interfacce utente del software in tutti i progetti?	3		
2.2	Metriche relative ai processi			
2.2.1	Sono mantenuti dati consuntivi sugli impegni delle risorse utilizzate nel progetto e poi confrontati con i dati di impegno pianificato?	2		
2.2.2	Sono mantenuti nel tempo i dati relativi alle dimensioni delle singole componenti software?	2		R
2.2.3	Sono mantenute statistiche sugli errori di disegno rilevati?	3		R
2.2.4	Sono mantenute statistiche sugli errori di codice rilevati?	2		R
2.2.5	Sono fatte previsioni sugli errori di disegno e poi paragonate con quelli a consuntivo rilevati?	4		
2.2.6	Sono fatte previsioni sugli errori di codice e poi paragonate con quelli a consuntivo rilevati?	4		R
2.2.7	Sono rilevati dati a consuntivo sui vari aspetti del codice e poi paragonati con quelli previsti?	2		
2.2.8	Sono rilevati i dati a consuntivo sui test unitari e poi paragonati con quelli previsti?	2		
2.2.9	Sono rilevati i dati a consuntivo per i componenti software integrati e poi confrontati con quelli pianificati?	2		
2.2.10	Sono rilevati i valori effettivi e quelli stimati di utilizzazione della	2		

Manuali di sviluppo di software

Num.	Domanda	Liv.	Si/ No	Rich.
	memoria dei computer destinatari del software sviluppato?			
2.2.11	Sono rilevati i valori effettivi e quelli stimati sul <i>throughput</i> dei computer utilizzati per il software sviluppato?	2		
2.2.12	Sono rilevati i valori di utilizzazione dei canali I/O?	2		
2.2.13	Sono misurati e registrati i livelli di copertura delle revisioni di disegno e codice?	4		R
2.2.14	Sono misurati e registrati i livelli di copertura dei test funzionali?	4		R
2.2.15	Sono tracciati gli errori rilevati durante le revisioni di disegno per poi verificarne la risoluzione?	3		R
2.2.16	Sono tracciati gli errori rilevati durante i test per poi verificarne la risoluzione?	2		
2.2.17	Sono tracciati i risultati delle revisioni di codice per poi verificarne la risoluzione?	3		R
2.2.18	E' tracciato lo stato di avanzamento dei test per ciascun prodotto software e confrontato con il piano?	2		
2.2.19	Sono mantenuti i dati relativi al contenuto del software rilasciato nel tempo?	2		
2.3 Analisi e gestione dei dati				
2.3.1	Il processo di sviluppo del software utilizzato è stabilito, gestito e controllato in base alle metriche utilizzate ed i dati relativi sono mantenuti in una base dati?	4		R
2.3.2	Sono analizzati i dati raccolti nelle revisioni di disegno?	4		R
2.3.3	Sono analizzati i dati raccolti nelle revisioni di codice e nei test, per determinare i tipi di distribuzione e le caratteristiche degli errori residui nel prodotto?	4		R
2.3.4	Sono condotte analisi degli errori rilevati per la determinazione della causa degli stessi?	4		R
2.3.5	E' utilizzato un meccanismo per effettuare l'analisi causale?	4		R
2.3.6	Sono riviste le cause degli errori per determinare le modifiche al processo richieste?	5		R
2.3.7	E' utilizzato un meccanismo per intraprendere azioni preventive per evitare gli errori?	5		R
2.3.8	E' analizzata l'efficacia delle revisioni per ciascun progetto?	4		R
2.3.9	E' analizzata la produttività per le attività principali del processo?	4		
2.4 Controllo dei processi				
2.4.1	Esiste un meccanismo che consente alla direzione di effettuare revisioni periodiche dello stato di avanzamento dei progetti di sviluppo software?	2		R
2.4.2	E' utilizzato un meccanismo per valutare periodicamente il	4		R

Manuali di sviluppo di software

Num.	Domanda	Liv.	Si/ No	Rich.
	processo di sviluppo software e realizzare i miglioramenti indicati?			
2.4.3	E' utilizzato un meccanismo per identificare e risolvere i problemi relativi allo sviluppo che hanno un impatto sul software?	3		
2.4.4	E' utilizzato un meccanismo per evidenziare al capo progetto i problemi rilevati durante i test?	3		
2.4.5	E' utilizzato un meccanismo per scambiare informazioni tecniche con il cliente?	2		
2.4.6	E' utilizzato un meccanismo che assicura l'aderenza agli standard di sviluppo software?	3		R
2.4.7	I responsabili dello sviluppo software (capi di prima linea) firmano le stime fatte sui tempi e sui costi?	2		R
2.4.8	E' utilizzato un meccanismo per assicurare la tracciabilità tra i requisiti software ed il disegno ad alto livello?	3		
2.4.9	E' utilizzato un meccanismo per controllare le modifiche ai requisiti software?	2		R
2.4.10	Esiste un processo formale gestionale per determinare se il prototipo di una funzione software è una parte del processo di disegno?	4		
2.4.11	E' utilizzato un meccanismo che assicuri la tracciabilità tra il software ad alto livello (macrofunzioni) ed il disegno di dettaglio?	3		
2.4.12	Sono eseguite revisioni tecniche interne sul disegno software?	3		R
2.4.13	E' usato un meccanismo per controllare le modifiche del disegno?	3		R
2.4.14	E' utilizzato un meccanismo per assicurare la tracciabilità tra il disegno di dettaglio ed il codice?	3		
2.4.15	Sono mantenute le registrazioni sullo stato di sviluppo dei moduli?	3		
2.4.16	Sono eseguite revisioni tecniche interne del codice?	3		R
2.4.17	E' utilizzato un meccanismo per controllare le modifiche al codice ovvero chi può modificare il codice e in quali circostanze?	2		R
2.4.18	E' utilizzato un qualche strumento per la gestione della configurazione del software?	3		
2.4.19	E' utilizzato un meccanismo per verificare che i campioni esaminati da SQA sono rappresentativi del lavoro eseguito?	3		R
2.4.20	E' utilizzato un meccanismo per assicurare che il test di regressione sono regolarmente eseguiti?	2		
2.4.21	E' utilizzato un meccanismo per assicurare l'adeguatezza dei test di regressione?	3		R
2.4.22	I casi di test sono revisionati?	3		

DOMANDE AGGIUNTIVE (Follow-up)

Sono fatte, quando opportuno, per espandere le risposte date alle domande base.

1. Durante la considerazione sull'assegnazione delle responsabilità, chiedere, come evidenza, alcuni rapporti/documenti in cui figuri il nome della persona, ecc...
2. Quando si considera un gruppo di lavoro, chiedere i loro nomi, l'organizzazione rappresentata e, come evidenza, qualche agenda di riunione in cui compaia il loro nome
3. Quando si considerano i programmi di formazione, chiedere un piano di corsi, date, contenuti dei corsi, nomi dei partecipanti, qualifica degli istruttori e degli studenti
4. Quando si considera l'esistenza di un meccanismo, procedura, standard, criteri o linee-guida, richiedere una copia del documento controllato, la storia delle modifiche/revisioni, i nomi dei responsabili, le loro job description, i rapporti recenti da loro emessi
5. Considerando l'utilizzo di profili, reportistica, paragoni tra dati effettivi e pianificati, misurazioni, richiedere gli ultimi tre rapporti in ordine temporale
6. Considerando il calcolo o l'analisi dei dati, richiedere gli ultimi tre rapporti che mostrino i risultati o le conclusioni raggiunte
7. Considerando lo svolgimento di attività, richiedere una copia di rapporti recenti che mostrino il tali attività
8. Considerando lo svolgimento di alcune azioni o utilizzo di strumenti, richiedere l'evidenza, sotto forma di procedura, responsabilità, sistema di tracciamento per dimostrare l'esecuzione
9. Considerando l'esistenza di uno strumento, pratica, metodo, richiedere come evidenza liste d'inventario, rapporti di utilizzo, manuali d'istruzione, piani di formazione, ecc...
10. Considerando l'uso di tool di automazione, richiedere una demo del tool.

TERMINOLOGIA

Procedura formale: un documento che descrive una sequenza di attività (fasi) con linee guida per l'utilizzo.

Meccanismo: strumento/mezzo/tecnica che assicuri l'efficacia di un'attività (task), un processo, una procedura. Il meccanismo può includere la combinazione di più elementi (piani operativi, descrizione di operazioni, procedure formali, strumento specifico, ecc.). La documentazione relativa definisce cosa occorre fare, come farla, chi è responsabile dei risultati.

Processo: una serie di meccanismi, attività (task), procedure a fronte un obiettivo. La documentazione relativa al processo di sviluppo software definisce la sequenza fasi da seguire per produrre un prodotto finito. Ciascun passo definito è descritto come una serie di attività (task) da svolgere usando una metodologia amministrativa e prescrive l'utilizzo di tecniche e strumenti (tool).

1.4.3 Rapporto di valutazione finale

Tabella 6. Rapporto di valutazione finale.

AZIENDA: PROCESSO: DATA VALUTAZIONE: PERSONE INTERVISTATE: VALUTATORI:	
ARGOMENTO	EVIDENZA IN AZIENDA
Obiettivi (la direzione ha definito obiettivi chiari e raggiungibili nell'ambito del miglioramento del processo in oggetto?)	
Impegno ad eseguire (la direzione e tutta l'organizzazione sono realmente impegnati ad eseguire il processo in oggetto così come definito?)	
Capacità di eseguire (la direzione ha definito un opportuno piano di formazione per assicurare le competenze necessarie ad eseguire il processo in oggetto così come definito?)	
Attività svolte (sono svolte le singole attività così come previsto dal processo in oggetto?)	
Misure ed analisi (sono definite le metriche e le misurazioni da eseguire relativamente al processo in oggetto? Sono eseguite analisi delle misurazioni effettuate?)	
Verifiche dell'applicazione (la direzione verifica periodicamente l'applicazione delle regole stabilite per seguire il processo in oggetto?)	
VALUTAZIONE FINALE (CONFORME/PARZIALMENTE CONFORME/NON CONFORME):	
NOTE:	
Firma valutatori	Firma intervistati

Nota: Allegare eventuali pagine aggiuntive a supporto di quanto riassunto nel rapporto.

Bibliografia

Di seguito è riportata la bibliografia che può essere utilizzata per approfondire i vari temi trattati. Resta, ovviamente, prioritaria la bibliografia relativa al SEI/CMM.

Capability Maturity Model

Software Engineering Institute (SEI) - 1997
The Capability Maturity Model – Guidelines for Improving the Software Process
Carnegy Mellon University

Software Engineering Institute (SEI)
Key practices of the Capability Maturity Model (CMM) Version 1.1
http://rbse.jsc.nasa.gov/process_maturity/CMM/TR25/

Normative ISO 9000 relative al software

ISO/IEC 9126: 1991, *Tecnologia dell'informazione – Valutazione del prodotto software – Caratteristiche qualitative e linee guida per la loro utilizzazione*

UNI EN ISO 9001: 1994, *Sistema qualità – Modello per l'assicurazione della qualità nella progettazione, sviluppo, fabbricazione, installazione e assistenza – Guida all'uso*

ISO 9000-3: 1998, *Linee guida per l'applicazione di ISO 9001 allo sviluppo, fornitura e manutenzione del software*

ISO/IEC 12207: 1995, *Tecnologia dell'informazione – Processi del ciclo di vita del software*