

Sviluppare software oggi ...

*La complessità di un'attività tecnica e creativa
divenuta indispensabile per la gestione del
business odierno: problemi e maturità del
mercato, cause e rimedi, soluzioni e speranza!*

Ercole Colonese

Versione 1.1 - giugno 2007

1

Note sulla versione attuale

Questa versione del documento riporta gli aggiornamenti portati sul sito a seguito di discussioni con il mondo del software, in particolare delle medie e piccole imprese di software italiane. Gli aggiornamenti nascono anche dall'esigenze di allineare quanto espresso nel sito con le esigenze del mondo della PA, in particolare con quanto espresso dalle Linee guida emesse dal CNIPA sul tema della qualità dei beni e servizi ICT.

Versioni del documento

2005	2006	2007	2008	2009
	V 1.0	V 1.1		

© Copyright Ercole F. Colonese 2006-2007

Ringraziamenti

Si ringraziano tutte le persone che titolo diverso hanno fornito suggerimenti ed evidenziato errori nel sito. Ringrazio in anticipo tutti coloro che vorranno inviare altri commenti e suggerimenti, segnalazioni di errori e mancanze, per migliorare ulteriormente il manuale.

Manuali di sviluppo software

Il presente manuale fa parte di una serie di documenti, in parte già disponibili ed in parte in fase di realizzazione, che mirano a fornire una vista dei vari temi dello sviluppo software. Essi riportano l'esperienza pratica maturata dall'autore in molti anni di lavoro. L'elenco dei manuali e del relativo stato di completamento riportato qui di seguito si riferisce alla data in cui è rilasciata la presente versione del manuale in oggetto.

1. Sviluppare software oggi ...	(disponibile)
2. Strategia di sviluppo e ciclo di vita del software	(disponibile)
3. Project Management	(disponibile)
4. Collaudo del software (testing)	(disponibile)
5. Collaudo del software in Rete	(in preparazione)
6. Qualità del software	(disponibile)
7. Modello ISO 9126 per la qualità del software	(disponibile)
8. Metriche del software	(in fase di completamento)
9. Usabilità del software	(disponibile)
10. Usabilità dei siti Web	(disponibile)
11. Metodi e tecniche dello sviluppo software	(in fase di completamento)
12. Introduzione al modello CMMI	(in fase di completamento)
13. Glossario dei termini dello sviluppo software	(disponibile)
14. Bibliografia	(disponibile)

I manuali completati sono disponibili nel sito dell'autore:
(<http://www.colonese.it/pubblicazioni/htm>).

A chi sono rivolti i manuali

I manuali sono rivolti a

- Studenti universitari che seguono i corsi di Ingegneria del software e che necessitano di approfondire i temi trattati;
- Capi progetto (Project Manager) che necessitano di una guida di riferimento per le attività di pianificazione e controllo delle attività di sviluppo del software;
- Sviluppatori, collaudatori, personale dell'assicurazione qualità e consulenti di processo che necessitano di approfondire i temi relativi al loro ruolo.

Sommario

Indice degli argomenti

1.	Sviluppare software oggi.....	9
1.1	Un'attività tecnica e creativa complessa.....	10
1.1.1	Ma è proprio così difficile?	11
1.1.2	Allora, cosa fare?	11
1.1.3	E l'esperienza cosa ci dice?	12
1.1.4	La metodologia proposta	13
1.1.5	La direzione aziendale prima di tutto	13
1.2	Problemi dei progetti software.....	14
1.2.1	Problemi gravi	14
1.2.2	Cause principali dei problemi del software	15
1.2.3	Alcune risposte	15
1.2.4	Come fare?	16
1.2.5	Un approccio metodologico organico, semplice ma efficace	16
1.2.6	Il ruolo della direzione	17
1.3	Necessità di migliorare per competere.....	17
1.4	Soluzioni ai problemi dei progetti software	19
1.4.1	ISO 9000.....	19
1.4.2	European Foundation for Quality Management (EFQM).....	20
1.4.3	Associazione Italiana Cultura Qualità (AICQ)	20
1.4.4	Centro Nazionale per l'Informatica nella Pubblica Amministrazione (CNIPA)	21
1.4.5	Project Management Institute (PMI)	21
1.4.6	Software Engineering Body of Knowledge (SWEBOK).....	21
1.4.7	Capability Maturity Model® Integration (CMMI®)	22
1.4.8	Software Process Improvement (SPI)	22
1.4.9	E la SA 8000?.....	22
1.4.10	Ma allora, ... cosa fare?.....	23
1.4.11	Maturità del mercato!.....	23
1.5	Maturità del mercato.....	23
1.5.1	Cosa può fare una piccola azienda?.....	23
1.5.2	E come si può fare?	24
1.5.3	Una conversazione tra addetti ai lavori	25
2.	Norme ISO 9000 nelle organizzazioni software	27
2.1	ISO 9000:2005, Sistemi di gestione per la qualità - Fondamenti e terminologia.	28
2.1.1	Contenuti della norma.....	28

2.1.2	Principi di gestione per la qualità.....	29
2.1.3	Scopo e campo di applicazione della norma.....	31
2.1.4	Motivazione alla base dei sistemi qualità.....	31
2.1.5	Requisiti per il sistema qualità e per i prodotti.....	32
2.1.6	Approccio dei sistemi qualità.....	32
2.1.7	Approccio per processi.....	33
2.1.8	Politica ed obiettivi per la qualità.....	34
2.1.9	Ruolo dell'alta direzione.....	34
2.1.10	Documentazione del sistema qualità.....	35
2.1.11	Valutazione dei sistemi di gestione per la qualità.....	36
2.1.12	Miglioramento continuo.....	37
2.1.13	Ruolo delle tecniche statistiche.....	38
2.1.14	Scopi dei sistemi qualità e di altri sistemi di gestione.....	39
2.1.15	Relazione tra i sistemi qualità ed altri sistemi di gestione.....	40
2.1.16	Termini e definizioni.....	40
2.2	ISO 9001:2000, Sistemi di gestione per la qualità - Requisiti.....	41
2.2.1	Contenuti della norma.....	41
2.2.2	Relazione con la ISO 9004.....	42
2.3	ISO 9004, Guida al miglioramento continuo.....	42
2.3.1	Contenuti della norma.....	42
2.4	ISO 19011, Guida alle verifiche ispettive.....	43
2.4.1	Contenuti della norma.....	43
2.5	ISO/IEC 9126, Modello per la qualità del software.....	44
2.5.1	Contenuti della norma.....	44
2.6	ISO 12207, Norme sul ciclo di vita del software.....	45
2.6.1	Una famiglia di norme per la gestione del ciclo di vita.....	45
2.6.2	Principali differenze con la ISO 9001.....	45
2.7	ISO 12207, Norme sul ciclo di vita del software.....	46
2.7.1	Contenuti della norma.....	46
2.7.2	Processi primari.....	47
2.7.3	Processi di supporto.....	48
2.7.4	Processi organizzativi.....	48
2.8	ISO/IEC TR 15271, Guida all'utilizzo della norma ISO/IEC 12207 sul ciclo di vita del software.....	49
2.8.1	Contenuti della norma.....	49
2.9	ISO/IEC TR 16326, Guida all'utilizzo della norma ISO/IEC 12207 nella gestione dei progetti software.....	49
2.9.1	Contenuti della norma.....	49
3.	Il modello CMMI nelle organizzazioni software.....	51
3.1	Livelli di maturità.....	52
3.1.1	Livello di maturità 1: Initial.....	52
3.1.2	Livello di maturità 2: Managed.....	53
3.1.3	Livello di maturità 3: Defined.....	53
3.1.4	Livello di maturità 4: Quantitatively Managed.....	54

3.1.5	Livello di maturità 5: Optimizing	54
3.2	Aree di processo (KPAs)	55
4.	La professione dell'ingegnere del software	57
4.1	Che cos'è la Software Engineering?	57
4.2	Che cos'è una professione riconosciuta?	58
4.3	Quali sono le caratteristiche di una professione?	58
4.4	Le competenze dell'ingegnere del software (SWEBOK).....	58
4.4.1	Le 10 aree di competenza (Knowledge Areas)	58
5.	Una metodologia a supporto dello sviluppo software	61
5.1	Competenze delle persone	63
5.1.1	Ruoli, responsabilità e competenze necessarie	63
5.1.2	SWEBOK	63
5.2	Processi maturi	64
5.2.1	Che cos'è un processo maturo?.....	64
5.2.2	Le fasi del ciclo di vita	65
5.2.3	La gestione del progetto	65
5.2.4	Livello di dettaglio	66
5.3	Utilizzo di metodi, tecniche, strumenti e metriche efficaci	66
5.3.1	Di cosa si tratta?	66
5.3.2	Metodi e tecniche a supporto dello sviluppo software	67
5.3.3	Metodi e tecniche	68
5.3.4	Tecnologie di sviluppo	71
5.4	Strumenti a supporto dello sviluppo.....	72
5.4.1	Strumenti di sviluppo	72
5.4.2	Ambienti di sviluppo	73
5.4.3	Supporto ai processi di sviluppo.....	73
5.5	Metriche del software	73
5.5.1	Perché misurare?	73
5.5.2	Cosa misurare?	74
5.5.3	Come misurare?	75
5.5.4	Alcuni esempi di metriche del software.....	75
Bibliografia		80
Ingegneria del software		80
Project Management.....		81
Testing.....		81
Metriche del software		82
Qualità.....		82
ISO9000.....		83
CNIPA.....		84
AICQ-CI.....		84

Indice delle figure

Figura 1. Modello di sistema di gestione per la qualità basato sui processi.	33
Figura 2. I cinque livelli di maturità del modello CMM.	52

Indice delle tabelle

Tabella 1. Differenze tra le norme ISO 9001 e ISO 12207.	46
Tabella 2. I processi associati ai cinque livelli di maturità del modello CMM.	56
Tabella 3. Obiettivi di un'organizzazione software.	75
Tabella 4. Dettagli sulle metriche relative agli obiettivi di un'organizzazione software.	76

1. Sviluppare software oggi...

Il manuale riporta quanto pubblicato sul sito dell'autore (www.colonese.it) sul tema dello sviluppo software. Il sito vuole scambiare opinioni sul modo IT, e su quello del software in particolare, divenuto troppo competitivo (purtroppo solo in termini economici) per riuscire a fornire il supporto allo sviluppo del sistema paese.

Il ruolo strategico che riveste l'IT (ed il software in tale contesto) nella crescita della competitività delle aziende e nel miglioramento dei servizi ai cittadini ed alle imprese è fin troppo evidente per essere qui ribadito.

Le aziende possono dare il meglio di loro stesse solo in un "mercato maturo". Un mercato dove la competizione porta a ridurre i costi senza scendere mai a compromessi con la qualità. Qualità intesa come "valore aggiunto". Come "supporto all'innovazione". Come erogazione di "servizi sempre migliori alla collettività ed ai singoli". Come "supporto alla crescita del sistema paese", appunto!

In tal senso le amministrazioni pubbliche giocano un ruolo importante: esse "acquistano qualità" dai fornitori per poi renderla "disponibile alla collettività". Qualità intesa come servizi di valore.

Migliorare la qualità del sistema paese è quindi l'unico fine plausibile per una crescita sostenibile del sistema stesso. Un circolo virtuoso dove i servizi migliorano la qualità della vita del cittadino e delle imprese. I prodotti innovativi aiutano le aziende a competere sul mercato. E i professionisti svolgono un ruolo di rilievo in tale contesto!

Lo sviluppo del software può e deve giocare un ruolo importante in tale senso, fornendo prodotti e servizi "di qualità" a "prezzi giusti".

Ma il mercato attuale sembra non essere così maturo da "valorizzare" in pieno la qualità. Pretenderla ma anche riconoscerla. Volerla a costi non realistici equivale a non ottenerla. E a soffrirne è, in definitiva, il sistema paese che stenta a trovare quella competitività che sicuramente sarebbe capace di affrontare!

In questo sito sono discussi molti temi legati al mondo dello sviluppo del software. Essi rispecchiano l'esperienza ed il punto di vista dell'autore.

Il tema dello sviluppo del software è trattato affrontando i seguenti temi:

- Un'attività creativa complessa
- Problemi dei progetti software
- Necessità di migliorare per competere
- Soluzioni ai problemi del software
- Maturità del mercato

Opinioni diverse e soluzioni alternative da quelle prospettate possono essere oggetto di discussione ed approfondimento sincero ed aperto, senza pregiudizi e senza condizionamenti di sorta.

1.1 Un'attività tecnica e creativa complessa

Lo sviluppo del software è un'attività tecnica e creativa ad alto contenuto umano.

L'evoluzione rapida delle tecnologie offre nuove possibilità di realizzazione, ma richiede un continuo aggiornamento delle competenze tecniche e la capacità di gestire la maggiore complessità.

I clienti chiedono ai propri fornitori "affidabilità, competenza e flessibilità" nell'affidare loro progetti di sviluppo, servizi di gestione del patrimonio applicativo.

Progetti e servizi pongono l'accento sul "rispetto dei tempi", il "contenimento dei costi" ed il "raggiungimento della qualità attesa".

Le aziende software, quindi, si trovano a competere in un ambiente sempre più difficile: sviluppare sistemi più complessi, in tempi più brevi ed a costi inferiori.

In questo scenario, i fornitori di prodotti e servizi sono chiamati ad operare con margini di profitto sempre minori e, qualche volta, "in perdita".

I progetti evidenziano ancora molti problemi dei progetti software: ritardo nelle consegne, costi oltre il budget, qualità inferiore alle attese!

Lo scenario non è tranquillizzante e l'andamento nel tempo lo è di meno!

Nasce quindi perentoria la necessità di migliorare per competere!

1.1.1 Ma è proprio così difficile?

L'ingegneria del software propone nuove metodologie, fornisce metodi e tecniche consolidate, segnala pratiche consolidate (best practice), ma...

... Molti progetti software sono afflitti da ritardi nelle consegne, costi oltre il budget, qualità inferiore alle attese!

Lo studio ha incentivato la ricerca di soluzioni ai problemi dei progetti software: prima fra tutte l'istituzione della branca "ingegneria del software"; quindi la costante ricerca di nuovi modelli di sviluppo e di miglioramento delle prestazioni per le organizzazioni software. Molti modelli sono diventati standard (ISO 9000, SA 8000, ecc.), altri sono diventati modelli per l'eccellenza (EFQM, ecc.), altri ancora modelli di maturità (CMMI, ecc.).

Molte organizzazioni hanno come missione la ricerca di soluzioni innovative per l'informatica in generale e per il mondo del software in particolare (Aicq, Cnipa, SEI, PMI, ecc.).

Cosa fare allora di tutti questi modelli? Quali adottare? E quanto costa adottarli?

Ma, ancora più importante: come far convivere questi modelli con le norme ISO 9000 diventate "cogenti" nel mercato europeo e quindi anche in quello italiano?

E' difficile ed oneroso affrontare un percorso di miglioramento da parte della piccola e media impresa per due ordini di ragioni. Il primo problema è rappresentato dalla scelta del modello più adatto alle proprie necessità (Quale modello è più adatto alle esigenze della mia azienda?). Il secondo è relativo all'onere richiesto dalla trasformazione, sia in termini finanziari che di coinvolgimento delle persone (Come faccio a coinvolgere le persone già così impegnate nelle attività quotidiane? Senza pensare che devo devo già affrontare le certificazioni ISO 9000!).

1.1.2 Allora, cosa fare?

Ciò che si richiede è che il mondo informatico nel suo complesso faccia un vero e proprio salto di qualità, faccia crescere il mercato in modo coerente e sostenibile:

- i clienti chiedano e pretendano soluzioni innovative, ma riconoscano il valore dei progetti; la PA chieda e pretenda soluzioni all'avanguardia, ma non propenda per offerte al minor prezzo (anche quando questi sono visibilmente non realistici!);
- i fornitori sappiano affrontare le sfide con professionalità e competenza;

- le associazioni aiutino e supportino materialmente le imprese a "crescere nella qualità".

In altre parole, è il mercato che deve crescere e il Sistema Italia fare il salto di qualità!

1.1.3 E l'esperienza cosa ci dice?

L'esperienza maturata in molti progetti di miglioramento dei processi produttivi e gestionali nell'ambito del software evidenzia la necessità di un approccio per gradi: crescere gradatamente ma con continuità, senza alcuna sosta né tentennamento!

Un primo elemento di criticità è rappresentato dal coinvolgimento della direzione, del management e del personale tecnico. Spesso la trasformazione è delegata ad un ristretto gruppo di persone, quando non ci si affida solo ai consulenti esterni, seguiti dall'addetto alla qualità dell'azienda. All'inadeguato coinvolgimento della direzione e del management al miglioramento fa spesso riscontro la preparazione del personale tecnico non adeguata ad affrontare le sfide da affrontare. I servizi, e tra questi lo sviluppo del software, basano la propria qualità proprio sulla competenza del personale!

Un secondo elemento di criticità è rappresentato dal processo di sviluppo software adottato. Ben documentato nei minimi particolari, esso è anche certificato ISO 9000 ma, nel suo utilizzo quotidiano nei progetti, è spesso disatteso perché poco adattabile alle specifiche circostanze dei progetti reali.

Un terzo ed ultimo elemento di criticità è rappresentato dallo scarso utilizzo di tecniche, metodi, strumenti e metriche a supporto delle attività quotidiane.

La soluzione giusta sembra essere quella che riesce a coniugare efficacemente i tre elementi sopra citati: competenza delle persone, processi maturi ed utilizzo di metodi, tecniche e strumenti adeguati alla complessità da affrontare.

Una metodologia, basata sui tre elementi cardini elencati, e che risulti semplice ed efficace, completa e rigorosa ma facilmente adattabile alle esigenze dell'azienda è quindi la "chiave del successo".

Un modello di metodologia di tale tipo è stato sperimentato con successo da un gruppo nutrito di piccole e medie imprese software italiane. I risultati sono stati confortanti ed il salto qualitativo evidente.

1.1.4 La metodologia proposta

La metodologia utilizza nel programma che ha visto coinvolte molte aziende di software poggia su tre pilastri irrinunciabili:

1. **Competenza delle persone** (elemento fondamentale di un'organizzazione);
2. **Processi maturi** per il software (in ottica ISO 9000 e CMMI);
3. **Metodi, tecniche, metriche e strumenti a supporto** ("best practice" prese in prestito dall'ingegneria del software e dai vari modelli di maturità e di eccellenza).

Ciascun elemento è calibrato per trarre il massimo vantaggio in termini di prestazioni e per limitare (quando non sia possibile annullare) ogni aspetto burocratico e formale.

Il modello recepisce le richieste del mercato, fa tesoro di quanto suggerito dalle varie linee guida emesse dagli organi competenti, utilizza quanto di meglio è disponibile. Indirizza le richieste espresse dalla Pubblica Amministrazione nei bandi di gara. Soddisfa i requisiti della norma ISO 90001:2000. Adotta le migliori pratiche consolidate e disponibili (alcuni elementi del modello di maturità CMMI, altre "best practices").

1.1.5 La direzione aziendale prima di tutto ...

L'esperienza dice che nessuno cambiamento significativo è possibile, nessun miglioramento è attuabile e nessun obiettivo è raggiungibile senza il coinvolgimento attivo della direzione e di tutto il personale.

La responsabilità della direzione nel progetto di miglioramento è l'elemento irrinunciabile per costruire il suo successo. A questo occorre aggiungere il coinvolgimento attivo di tutto il personale che tale cambiamento dovrà "farlo suo" e "renderlo operativo".

Entrambi questi elementi sono il presupposto di ogni programma di miglioramento.

Nessuna azienda può crescere se la direzione non ne è veramente convinta, non diventa leader del cambiamento e non riesce a coinvolgere il personale nella trasformazione!

1.2 Problemi dei progetti software

Lo sviluppo del software è un'attività creativa ad alto contenuto umano e quindi soggetta ad errori.

Errori ed incomprensioni nelle finalità (obiettivi, requisiti, standard, ecc.) fanno del software un oggetto molto instabile, costoso e spesso non adeguato alle necessità dell'utilizzatore finale.

La presenza di errori nel software causa malfunzionamenti che devono essere corretti. E questo ha un costo. Gli interventi sul software diventano più onerosi al crescere della sua complessità e quanto più tardi sono rilevati gli errori: più complesso è il codice e più costoso sarà modificarlo, più tardi scopriamo gli errori e più ci costerà risolverli!

Ma il codice è modificato anche per potenziarne le funzionalità, adeguarlo a normative e standard, migliorarne le prestazioni. Si tratta di un'operazione non facile: innestare nuovo codice in un software non progettato per essere modificato, a volte troppo complesso e poco strutturato, è un'operazione difficile e costosa.

Sviluppare nuove applicazioni è il sogno di tutti noi. Realizzare qualcosa partendo da zero. Senza dover intervenire su cose fatte da altri (e fatte male, come sempre le cose fatte dagli altri!).

Ma anche in questi casi, i risultati non cambiano ed i problemi rimangono gli stessi: ritardo nelle consegne, costi oltre il budget, qualità inferiore alle attese!

Molti progetti falliscono o non raggiungono i risultati attesi.

Perché?

1.2.1 Problemi gravi ...

La disciplina dell'ingegneria del software, benché ricca di studi e ricerca, è ancora poco applicata nella pratica quotidiana. A differenza di altre branche dell'ingegneria, nel campo del software si "naviga a vista", invece che con l'ausilio di strumentazione. Si produce "manualmente", piuttosto che con strumenti di produttività. Si "spera" in risultati positivi, e non si controlla con metodicità.

In estrema sintesi, ci si avvale di competenze approssimative e non di professionisti del software.

Molti studi sono stati condotti per capire le cause del fallimento di tantissimi progetti software.

I risultati delle analisi hanno sempre evidenziato problemi gravi nelle aree:

- interpretazione dei requisiti;
- pianificazione dei progetti;
- collaudo dei prodotti;
- attenzione alla qualità.

1.2.2 Cause principali dei problemi del software ...

Le cause principali sono state identificate in:

- poca attenzione dedicata all'analisi dei requisiti;
- bassa qualità della progettazione;
- stima poca realistica delle dimensioni dei progetti;
- pianificazione "ottimistica" dei progetti o condizionata dalle sole aspettative della direzione;
- risorse non adeguate alla complessità del progetto (come numero e come competenza);
- assenza di revisioni tecniche dei prodotti intermedi realizzati;
- inadeguatezza dei collaudi eseguiti;
- scarso controllo della qualità.

1.2.3 Alcune risposte ...

L'ingegneria del software è nata proprio per risolvere "la crisi del software", così come fu enunciato nel convegno NATO a Garmisch, in Germania, nel 1968.

Essa studia i problemi relativi allo sviluppo del software e ricerca metodi, tecniche e strumenti per affrontarli in modo sistematico. Le migliori fra queste e che dimostrano sul campo la loro efficacia sono dette "best practice".

Le norme ISO9000 sono state adeguate (Vision 2000) per fornire un modello di riferimento più maturo e adeguato al contesto attuale, in continua evoluzione e competitività crescente.

Il CMM®I (Capability Maturity Model®, Integration) fornisce anch'esso un modello efficace basato sul livello di maturità dei singoli processi (KPA - Key Process Area).

Il PMBOK (Project Management Body Of Knowledge) fornisce la base di conoscenza per una corretta gestione dei progetti, specialmente di quelli software.

Il SWEBOK (Software Engineering Body Of Knowledge) fornisce invece la base di conoscenza per i professionisti del software (gli ingegneri del software).

1.2.4 Come fare?

I modelli quindi esistono e, forse, sono anche troppi (meglio troppi che pochi!). Il problema, semmai, è quali adoperare e come applicarli in modo efficace compatibilmente con le attività quotidiane. Gli impegni di lavoro devono comunque essere onorati. Ed il cambiamento richiede tempo e denaro.

Facile a dirsi, meno facile a farsi!

Occorre la strategia dei piccoli passi. Piccoli ma continuativi, senza mai fermarsi. Poco ma ogni giorno, inesorabilmente.

Porsi pochi obiettivi per volta, "raggiungibili" e "in linea con il business quotidiano". E poi perseguirli con rigore (quasi con mania!).

E poi misurare i risultati, anche se non eclatanti, ed insistere nel miglioramento. Condividere i risultati con tutta l'organizzazione (perché i risultati sono sempre il frutto dell'intera organizzazione).

Quindi porsi un altro obiettivo raggiungibile ed insistere nel perseguirlo.

"Anche un lungo viaggio inizia sempre con un primo passo" (ha detto qualcuno molto saggiamente ed io lo riporto qui).

Tutto ciò vuol dire che non esistono rimedi miracolosi, ma solo "tanto olio di gomito" (altra frase presa in prestito).

1.2.5 Un approccio metodologico organico, semplice ma efficace

Un approccio metodologico organico, semplice ma efficace, testato con successo sul campo costituisce un aiuto.

Una soluzione di questo tipo è, per esempio, quella di affrontare un percorso di miglioramento dell'organizzazione che includa con pari efficacia tutti e tre gli aspetti più volte menzionati: le competenze delle persone, sia tecniche che manageriali, i processi produttivi e gestionali; l'utilizzo di metodi, tecniche, metriche e strumenti a supporto delle attività.

Una metodologia di questo tipo è stata sperimentata con successo in progetti che hanno visto coinvolto un grande numero di piccole e medie imprese di software italiane.

Già definita nelle sue linee principali, essa poggia su tre pilastri irrinunciabili:

1. la **competenza** delle persone (competenze tecniche e manageriali);
2. i **processi** maturi per il software (in ottica di miglioramento continuo);
3. i **metodi, le tecniche, le metriche e gli strumenti** a supporto delle attività produttive e gestionali.

Il tutto calibrato in modo da costituire un "unicum" equilibrato di tecniche e competenze, metodi e strumenti, atteggiamento positivo e metriche per misurare i risultati conseguiti.

1.2.6 Il ruolo della direzione ...

Per il successo di una tale iniziativa occorre il coinvolgimento attivo della direzione aziendale.

Essa gioca un ruolo importante nel cambiamento: leader della trasformazione, motore e ispiratore del miglioramento.

Non si è mai vista un'organizzazione migliorare le proprie prestazioni senza che la direzione ne veda la necessità, ne capisca l'importanza, si ponga alla testa del cambiamento, ne faciliti le azioni e ne valuti i progressi.

1.3 Necessità di migliorare per competere

Il software è sempre più presente nella vita quotidiana. E' lo strumento con cui le aziende automatizzano i propri processi per operare in un mercato globale sempre più competitivo. Si richiedono soluzioni software in tempi brevi, a costi inferiori e con qualità superiore. E non sempre ciò è fattibile per i fornitori di soluzioni software. Ci si prova, anzi, bisogna provarci, ma non sempre ci si riesce. In altre parole, il software è uno strumento di grande aiuto per risolvere i problemi di competitività delle aziende. Migliorare le competitività significa aumentare la produttività e migliorare la qualità dei prodotti e servizi realizzati.

In altre parole, significa agire sui seguenti elementi:

- produrre a costi inferiori (cioè a costi competitivi);
- ridurre i tempi di realizzazione (cioè rilasciare i prodotti nei tempi richiesti dal mercato e dal cliente);
- migliorare la qualità dei prodotti realizzati (cioè essere aderenti ai requisiti richiesti dal mercato e dal cliente).

La realizzazione di un Sistema per la gestione della qualità (SGQ) secondo le norme ISO 9000 rappresenta uno strumento potente ed efficace per un'azienda che voglia operare in un contesto difficile ed in continua evoluzione come quello del software. La competitività nel mercato globale richiede un sistema di processi in continua evoluzione con un miglioramento continuo in termini di efficacia ed efficienza.

Il nuovo quadro normativo ISO 9000 del 2000 rappresenta un modello standard e sicuro.

Da numerosi analisi eseguite da esperti del settore (vedi, per esempio, gli articoli pubblicati dall'Associazione italiana per la cultura della qualità, sottocomitato per il software, www.aicqci.it) risulta che le organizzazioni software certificate è ancora solo una parte del numero totale di aziende del settore. Di queste, poi, una parte ancora minore utilizza il proprio Sistema per la gestione della qualità come vero strumento per il miglioramento continuo. I processi produttivi non sono ancora misurati, monitorati e valutati in maniera sistematica e migliorati in termini di efficacia ed efficienza.

La certificazione ISO 9000 è ancora utilizzata da molti (per fortuna non da tutti!) come qualcosa da esibire piuttosto che come strumento reale di miglioramento e quindi di competitività!

In questo quadro non sono esenti da colpe i clienti che spingono molto sui prezzi e poco sulla qualità!

Ci sono deboli segnali nella direzione opposta, per fortuna, specialmente nell'ambito della Pubblica Amministrazione.

Speriamo che tali segnali diventino presto una realtà concreta su cui impostare la rinascita del settore!

Ma vediamo ora come le attuali norme ISO 9000 possono aiutare le aziende di software a migliorare le performance.

1.4 Soluzioni ai problemi dei progetti software

I problemi dei progetti software sembrano essere diventati cronici.

Molti, troppi progetti sono ancora afflitti dai soliti problemi: ritardi nelle consegne, costi oltre il budget, qualità inferiore alle attese!

Gli stessi problemi che affliggono i progetti software da più di quaranta anni. La ricerca di soluzioni appropriate va avanti da moltissimo, ma i problemi si permangono.

Come prima soluzione (eravamo nel 1968) si è creata la nuova disciplina dell' "Ingegneria del software". Le nuove tecnologie offrono potenziali di sviluppo notevoli. Ma i problemi del software non cambiano.

La ricerca di nuovi modelli di miglioramento delle prestazioni delle organizzazioni software è un cammino parallelo che prosegue con impegno.

Alcuni modelli sono diventati degli standard (ISO 9000), altri sono diventati modelli di eccellenza (EFQM), altri ancora modelli di maturità (CMMI), altri infine modelli di riferimento per le competenze (PMBOK, SWEBOK). E altri modelli ancora che non conosco.

Molte istituzioni (Aicq, Cnipa, PMI, SEI, ecc.) operano per diffondere la cultura della qualità, stabilire regole, fornire linee guida, proporre metodologie consolidate, preparare i professionisti. Tutti con lo stesso fine ultimo: aiutare il mercato ad affrontare i problemi con la giusta attenzione, crescere in maniera coerente, competere sul mercato sempre più esigente.

Ma cosa propongono tutti questi modelli? Quali vantaggi concreti garantiscono? E perchè sono tanti e non uno solo, magari flessibile per indirizzare necessità diverse? E poi. Sono compatibili tra di loro oppure in alternativa, o peggio in contrasto?

I ultima ed estrema analisi: sono la soluzione che stiamo cercando o un ulteriore elemento di confusione per chi deve lottare, giorno per giorno, per la sopravvivenza?

Proviamo a guardarci dentro, solo un poco, giusto per capirne di più.

1.4.1 ISO 9000

Le norme ISO 9000 sono state adeguate nel 2000 (Vision 2000) per fornire un modello di riferimento più evoluto ed aderente al contesto attuale. Ma i sistemi di gestione per la qualità, realizzati e certificati

secondo le norme, sembrano non aiutare concretamente le aziende a risolvere i loro problemi di performance. Perché?

In questo ambito l'AICQ-CI (Associazione Italiana per la Cultura della Qualità - Centro Insulare) opera con grande impegno per diffondere la cultura della qualità.

In particolare, il Comitato Software organizza eventi specifici (seminari, corsi, pomeriggi culturali, ecc.) sui temi più caldi. Pubblica una serie di Quaderni dedicati alle piccole e medie imprese sui vari temi del software. Tra questi, una utilissima "Guida all'adozione delle norme ISO 9000:2001 nelle organizzazioni software".

Ma tutto questo sembra non bastare!

1.4.2 European Foundation for Quality Management (EFQM)

Il Modello EFQM (European Foundation for Quality Management) è nato con l'obiettivo di supportare il miglioramento delle prestazioni aziendali attraverso un approccio complessivo più esteso ed articolato rispetto ai modelli classici ISO 9000.

Il modello consente alle aziende di autovalutarsi e di partecipare all'EFQM Excellence Award (EQA), un premio gestito dalla stessa fondazione (EFQM).

Il modello valuta gli elementi adoperati dall'azienda (leadership, gestione del personale, politiche e strategie, gestione delle risorse, processi) in base ai risultati ottenuti (soddisfazione del personale e del cliente, impatto sulla società, risultati di business).

E' un ottimo modello, anche per le piccole e medie imprese, ma costituisce un ulteriore fattore di investimento, non sempre alla portata di tutti!

1.4.3 Associazione Italiana Cultura Qualità (AICQ)

L'Aicq è un'associazione "no profit" con l'obiettivo di contribuire alla diffusione della cultura della qualità nel nostro Paese.

Al suo interno opera un "comitato per la qualità del software" (Aicq-ci) che si dedica allo studio ed alla diffusione delle problematiche relative al mondo software. Il comitato, forte dell'esperienza dei suoi membri (maturata nel mondo produttivo informatico), organizza eventi, tiene seminari e corsi di aggiornamento, pubblica quaderni ed una rivista trimestrale, ecc.

Sull'argomento specifico qui discusso, ha realizzato un utilissimo quaderno con le "Linee guida per l'adozione della norma ISO 9001:2000 nelle organizzazioni software".

Una goccia d'acqua nell'oceano del mercato della competitività. Ma è sempre meglio di niente!

1.4.4 Centro Nazionale per l'Informatica nella Pubblica Amministrazione (CNIPA)

Il CNIPA organizza moltissimi eventi sui vari temi dell'ingegneria del software e sulla gestione dei contratti IT nella Pubblica Amministrazione (PA).

In particolare, ha realizzato una collana di Quaderni sulla qualità delle forniture IT nella PA: "Linee guida" per indirizzare la qualità dei beni e servizi ICT nella PA fornendo una descrizione semplice e completa delle modalità contrattuali richieste dalla PA e come indirizzarle con efficacia da parte dei fornitori.

Ma ancora molti contratti IT nella PA soffrono dei problemi denominati prima come "cornici": ritardi, extracosti, qualità inadeguata!

1.4.5 Project Management Institute (PMI)

I "Chapter" italiani del Project Management Institute (PMI) si sono adoperati per diffondere la cultura della gestione dei progetti con azioni formative e di certificazione. Ora il manuale PMBOK (Project Management Body of Knowledge) è disponibile anche in lingua italiana!

1.4.6 Software Engineering Body of Knowledge (SWEBOK)

La competenza delle persone addette al settore dell'informatica rappresenta una chiave di grande importanza per il successo di un progetto software. L'argomento è stato affrontato in maniera sistematica con uno specifico modello, risultato di un lavoro di ricerca da parte del SEI e di rappresentanti di molte università e mondo produttivo internazionale.

Il modello è descritto nel documento "Software Engineering Body of Knowledge" (SWEBOK).

Nato negli USA, è ancora poco utilizzato dalle aziende.

In Italia, specialmente nelle università dove si formano i futuri professionisti del software, è poco conosciuto ed ancor di meno utilizzato!

1.4.7 Capability Maturity Model® Integration (CMMI®)

Il modello di maturità CMMI® rappresenta un modello molto diffuso, specialmente fuori dall'Italia, per il miglioramento delle prestazioni aziendali.

E' basato su livelli di maturità crescenti ciascuno dei quali indirizza singole aree di processo (Gestione dei requisiti, Gestione dei progetti, Verifica e collaudo del software, Gestione della configurazione, ecc.).

Il Software Engineering Institute (SEI) garantisce una continua evoluzione del modello tenendo conto dei riscontri relativi all'applicazione del modello stesso.

L'adozione del modello CMMI nelle organizzazioni software ha incontrato un certo successo negli USA dove il modello è nato su esplicita richiesta del Dipartimento della Difesa Americana.

In Europa, ed in particolare in Italia, il modello non è ancora molto diffuso. Forse perchè è percepito come troppo complesso? Forse richiede investimenti non proponibili per le imprese italiane, specialmente quelle di dimensioni medio piccole? Forse perchè richiede uno sforzo ulteriore rispetto alla certificazione ISO 9000, comunque necessaria per competere nel mercato europeo ed italiano?

1.4.8 Software Process Improvement (SPI)

L'area del "Software Process Improvement" (SPI) si è arricchita di molti altri modelli di maturità, proliferati negli ultimi anni e che indirizzano specifiche aree di interesse (le competenze, il testing, la progettazione, la documentazione, ecc.).

Tutti focalizzati al miglioramento continuo dei processi di sviluppo del software, questi modelli creano forse "confusione" in chi ha problemi di sopravvivenza ed ha poco tempo, risorse ed energia per affrontare il problema in maniera strutturata!

1.4.9 E la SA 8000?

Si tratta di una norma internazionale che regola alcuni aspetti della gestione aziendale attinenti alla responsabilità sociale d'impresa (SA 8000 sta per Social Accountability).

1.4.10 Ma allora, ... cosa fare?

Questi modelli sembrano essere tutti buoni. Ottimi in determinate situazioni ma difficili da implementare. E allora, cosa posso fare?

1.4.11 Maturità del mercato!

Credo fermamente che la soluzione stia nella crescita del mercato in termini di maturità. L'argomento è discusso in un successivo paragrafo.

1.5 Maturità del mercato

Credo fermamente che la soluzione dei problemi delle aziende che operano nel settore IT (e software in particolare) stia nella crescita del mercato in termini di maturità.

Maturità del mercato intesa come maturità del Sistema Paese. Sistema dove le componenti principali (Istituzioni, Mondo produttivo, Collettività) vivono e crescono in un confronto continuo e dialettico, ma in armonia con il contesto generale. In armonica simbiosi, direi. Perché ognuna ha bisogno delle altre. Per vivere e crescere. Parlo di un sistema dove l'interesse del singolo deve fare sempre i conti con quello della collettività. E dove la collettività rispetta il singolo. Dove le regole sono rispettate. Da tutti.

Allora si crea un circolo virtuoso dove la qualità è valore. E' il metro con cui si valuta anche l'adeguatezza del prezzo. Risultato: vince chi produce valore a prezzi competitivi. E' sempre la collettività (ed il suo bene) il fine ultimo. Cittadini e aziende usufruiscono di servizi migliori, più adeguati alle esigenze, capaci di migliorare la qualità della vita e semplificare la burocrazia per le aziende. In tutto questo le Istituzioni pubbliche fanno da elemento trainante e moderatore allo stesso tempo. Incentivano l'innovazione, la competitività, la produttività ... Indirizzano gli sforzi e gli investimenti per creare condizioni di maggiore benessere e competitività.

Come dire? Ognuno gioca il proprio ruolo, in sinergia ed in competizione con gli altri. In un contesto sano e maturo: la crescita del Sistema Paese.

1.5.1 Cosa può fare una piccola azienda?

Per un'organizzazione di piccole o medie dimensioni la sfida è ancora maggiore ed il percorso da compiere più arduo. Ma cosa fare?

Cambiare, rinnovare, migliorare è l'unica via percorribile che io conosca.

Migliorare per competere (e vincere) è la chiave del successo. Il miglioramento continuo, l'approccio da seguire.

Miglioramento continuo. Anche a piccoli passi, ma senza mai fermarsi. Con caparbia, insistenza, quasi con maniacale perseveranza.

Anche un lungo viaggio (l'ho già detto) inizia con un primo passo e continua con ... tanti altri passi, in sequenza. Tutti verso la stessa direzione, per raggiungere la meta.

1.5.2 E come si può fare?

Un cambiamento così radicale nella cultura di un'azienda non è mai facile da effettuare.

Si tratta di un processo di cambiamento che richiede elementi tanto importanti quanto difficili da trovare, almeno tutti insieme.

Ingredienti? Convinzione e leadership; Innovazione e creatività; Professionalità e passione.

Convinzione e leadership.

Parlo di un management effettivamente convinto della necessità di cambiare e di migliorare per competere e vincere. Solo se convinto realmente può esercitare la leadership che ogni cambiamento richiede. Vincere le resistenze e condurre l'organizzazione verso nuovi modelli, più efficaci ed efficienti. In grado di assicurare competitività e successo.

Innovazione e creatività.

Parlo anche di soluzioni semplici ed efficaci, adatte al contesto in gioco (e non belle solo in teoria!). Innovazione e creatività nascono anche dall'analisi del quotidiano affrontando i problemi in modo diverso da quello abituale. La soluzione è quasi sempre già presente, vicina a noi. "Geniale nella sua semplicità" diciamo quando la soluzione vincente è di un altro. "L'uovo di Colombo. Perché non ci ho pensato io prima?". Perché troppo presi dal problema in sé; giustamente preoccupati dalle possibili conseguenze del problema stesso.

Immaginiamo di porci domande di questo tipo in una riunione con le persone giuste in azienda, quelle che ci credono (quelli che poi fanno le cose!):

"Ma il cliente di cosa ha veramente bisogno?"

"Se fossi noi al suo posto, cosa vorremmo veramente?"

"Veramente si ha bisogno di tutto e subito?"

"Cosa serve prima e cosa dopo e cosa magari mai?"

"Siamo sicuri che la cosa si possa fare nell'unico modo in cui si è sempre fatto?"

"Come si può fare bene quella cosa? E perchè noi non possiamo farla così?"

"Possiamo farla ugualmente bene in un altro modo? E allora perchè non la facciamo?"

... ed altre domande ancora (che già vi sento fare).

Professionalità e passione.

Parlo quindi di persone competenti, motivate, capaci di sacrifici. Persone in grado di guardare oltre quando è il momento di pensare. Di mettersi in gioco quando si deve decidere. E fare bene quando si deve realizzare. Parlo di "professionisti".

Ma dove sono finiti quelli veri? Dove si possono trovare? E dove si creano quelli nuovi? Dove? Allora cerchiamoli! E sappiamoli trattenere. Sono il vero (unico) valore di un'azienda!

1.5.3 Una conversazione tra addetti ai lavori ...

Amici che hanno aziende di software mi chiedono: "Conosci qualche bravo programmatore? Uno di quelli veramente bravi, che ha passione per il lavoro che fa. Non importa quale linguaggio di programmazione conosca. L'importante è che sia bravo. Un nuovo linguaggio lo si impara presto. A lavorare bene no!".

Risposta: "Mi dispiace. Ho perso l'indirizzo dove li costruiscono. Non so proprio dove trovarlo!".

Fine della conversazione: "Peccato! Ne avevo proprio bisogno. Ho una bella idea da realizzare e non trovo le persone giuste!".

2. Norme ISO 9000 nelle organizzazioni software

Lo standard ISO 9000 attuale (Vision 2000) ha compiuto un passo notevole verso il miglioramento continuo della qualità. Un'azienda software trova ora quanto necessario per portare l'organizzazione verso un miglioramento concreto e tangibile, ma ...

Occorre evitare di costruire cattedrali nel deserto, sistemi di qualità perfettamente aderenti alle norme ma poco utili dal punto di vista produttivo. Sistemi realizzati solo per risultare "conformi" (solo per ottenere il "bollino blu") sono molto costosi e non producono risultati economici che giustifichino gli investimenti fatti. Occorre andare alla sostanza, semplificare, condividere e poi...

AGIRE. Con competenza, continuità e coerenza!

L'Aicq-ci (Associazione italiana per la cultura della qualità - centro insulare) - sottocomitato per la qualità del software, a cui l'autore si onora di appartenere - ha realizzato un utilissimo quaderno per le organizzazioni software che descrive l'interpretazione delle norme e si suggeriscono modalità per la loro applicazione:

"Linee Guida per l'applicazione della norma ISO 9001:2000 (Vision 2000) per lo sviluppo e la manutenzione del software - Quaderno N. 11 - Aprile 2001".

Il quadro normativo attuale è stato notevolmente semplificato ed è costituito da quattro documenti principali:

- **ISO 9000** - Fondamenti e terminologia, descrive i fondamenti dei sistemi di gestione per la qualità e ne specifica la terminologia adoperata;
- **ISO 9001** - Requisiti per i sistemi di gestione per la qualità, specifica i requisiti dei sistemi di gestione per la qualità che un'azienda deve soddisfare per dimostrare ai propri clienti, ed al mercato in generale, la propria capacità di fornire prodotti e servizi in linea con la qualità attesa;
- **ISO 9004** - Guida al miglioramento continuo, fornisce linee guida per adottare le norme ISO 9000 in maniera proficua, migliorando le prestazioni aziendali, sia in termini di efficacia che di efficienza, in modo continuativo (miglioramento continuo);

- **ISO 19011** - Guida alle verifiche ispettive dei sistemi di gestione per la qualità ed ambientali, fornisce una guida per le verifiche ispettive da condurre su sistemi di gestione per la qualità ed ambientali.

A queste si aggiungono norme specifiche per lo sviluppo del software, tra cui le più significative:

- **ISO 9126**, Le norme sulla qualità dei prodotti software;
- **ISO 12207**, Le norme sul ciclo di vita del software

Approfondire ciascun documento selezionandolo dalla lista.

2.1 ISO 9000:2005, Sistemi di gestione per la qualità - Fondamenti e terminologia.

2.1.1 Contenuti della norma

"La norma fornisce i principi e i concetti fondamentali di sistemi di gestione per la qualità su cui si basano le norme ISO 9000. Essa include i termini e le definizioni riguardanti gli stessi sistemi di gestione.

La norma si applica in tutte le organizzazioni indipendentemente dal tipo e dimensione e dai prodotti forniti e in tutte le attività interne ed esterne alle organizzazioni per l'elaborazione di documenti, norme e specifiche alla qualità." (definizione tratta dalla norma stessa).

Qui di seguito si fornisce una breve descrizione dei principi fondamentali dei sistemi qualità per fornire al lettore una idea di massima. Per una visione completa e dettagliata occorre fare riferimento di rettamente alla norma in questione. Gli argomenti sono esposti con la seguente struttura:

- Fondamenti dei sistemi qualità
- Termini e definizioni

Fondamenti dei sistemi qualità

I fondamenti dei sistemi qualità prevedono i seguenti elementi di base, ciascuno descritto brevemente nel seguito:

- Principi di gestione per la qualità;
- Scopo e campo di applicazione della norma;
- Motivazione alla base del sistema qualità;

- Requisiti per il sistema qualità e per i prodotti;
- Approccio dei sistemi qualità;
- Approccio per processi;
- Politica ed obiettivi per la qualità;
- Ruolo dell'alta direzione;
- Documentazione del sistema qualità;
- Valutazione del sistema qualità;
- Miglioramento continuo;
- Ruolo delle tecniche statistiche;
- Scopi dei sistemi qualità e di altri sistemi di gestione;
- Relazione tra i sistemi qualità ed i modelli di eccellenza.

Termini e definizioni

La norma riporta le definizioni dei termini utilizzati, raggruppati nelle seguenti tipologie: termini relativi alla qualità, termini relativi alla gestione, ecc.

2.1.2 Principi di gestione per la qualità

Guidare e far funzionare con successo un'organizzazione significa dirigerla e tenerla sotto controllo in maniera sistematica e trasparente.

Il successo di un'azienda deriva dall'attuazione di politiche mirate che pongono un'attenzione continua verso le esigenze dei propri clienti, dipende dall'utilizzo di processi ottimizzati, è fortemente condizionata dalla competenza e motivazione del personale tecnico e dirigenziale coinvolto, richiede un continuo monitoraggio delle prestazioni fornite e necessità che siano prese azioni correttive e preventive efficaci e di rapida attuazione.

Ultimo ingrediente, forse di maggiore rilevanza, il successo richiede che l'alta direzione diventi "leader" del cambiamento necessario, sia il promotore delle azioni di miglioramento, sia di esempio per tutti nell'affrontare le sfide a viso aperto, crei un clima di fiducia e di coinvolgimento positivo che riconosca i giusti meriti del personale, che garantisca la crescita professionale necessaria ad affrontare le sfide del mercato, che pretenda infine i risultati attesi.

Le nuove norme ISO 9000 hanno definito otto principi per una "corretta" gestione dei sistemi qualità:

1. **Orientamento al cliente.** Le aziende dipendono dai propri clienti e quindi devono capire le loro esigenze presenti e future, soddisfare i loro requisiti e mirare a superare, quando possibile, le loro stesse aspettative.
2. **Leadership.** I capi (il management a tutti i livelli) devono stabilire obiettivi in linea con la politica aziendale, raggiungibili dall'organizzazione e fornire le risorse necessarie per il loro raggiungimento. Devono altresì creare e mantenere un clima interno sereno, motivante che veda il pieno coinvolgimento di tutto il personale aziendale nel raggiungimento degli obiettivi stabiliti.
3. **Coinvolgimento del personale.** Le persone rappresentano il patrimonio più ricco e importante per un'azienda. La loro professionalità e motivazione, il loro coinvolgimento attivo sono l'elemento chiave per affrontare le sfide e raggiungere gli obiettivi stabiliti. Occorre creare e mantenere un clima positivo che permetta al personale di mettere a disposizione dell'azienda le loro capacità. La crescita continua della professionalità è necessaria per affrontare con successo il mercato. La gestione del personale in tale ottica è vera sfida per tutto il management!
4. **Approccio basato sui processi.** Occorre pianificare, svolgere e controllare le attività secondo schemi e modelli che definiscano le modalità operative e permettano di predirne il risultato. Tale modo di operare si dice "approccio basato sui processi".
5. **Approccio sistemico alla gestione.** Identificare, capire e gestire i processi tenendo conto delle loro correlazioni e vederli come se fossero parte di un "sistema" contribuisce a migliorare l'efficacia dell'organizzazione e permette di ottimizzarne l'efficienza. Risultato: maggiore confidenza nella capacità di raggiungere gli obiettivi!
6. **Miglioramento continuo.** Il miglioramento continuo delle prestazioni aziendali deve essere un obiettivo permanente dell'organizzazione. Il mercato e la concorrenza sono in continua evoluzione. Non ci si può fermare nemmeno sugli allori. Mai.
7. **Decisioni basate su dati di fatto.** Le decisioni per essere efficaci si devono basare su dati di fatto, su informazioni attendibili ed aggiornate. L'istinto e l'intuito aiutano, l'emotività no!
8. **Rapporto di reciproco beneficio con i fornitori.** L'azienda ed i suoi fornitori sono interdipendenti: il business di uno è legato al business dell'altro. Il successo di uno dipende anche dal successo

dell'altro. La collaborazione e la reciproca stima sono elementi imprescindibili ed i benefici sono evidenti per entrambe le parti.

2.1.3 Scopo e campo di applicazione della norma

La ISO 9000 è una norma internazionale e descrive i fondamenti dei sistemi di gestione per la qualità. Essa si applica a:

- Organizzazioni che vogliono ottenere vantaggi tangibili alle loro performance aziendali applicando un sistema di gestione per la qualità;
- Organizzazioni che necessitino la garanzia da parte dei propri fornitori circa i prodotti ed i servizi forniti;
- Chiunque utilizzi un prodotto e voglia garanzia circa la sua qualità;
- Chi sia interessato ad avere una comune comprensione della terminologia utilizzata nella gestione per la qualità (aziende, fornitori, clienti, consulenti qualità, ecc.);
- Coloro, interni od esterni all'azienda, che valutano i sistemi di gestione per la qualità o ne verificano la conformità alle norme ISO 9000;
- Chiunque altro sia interessato a conoscere le norme per motivi diversi.

2.1.4 Motivazione alla base dei sistemi qualità

Un sistema qualità, se ben fatto, può aiutare un'azienda a migliorare, anche sensibilmente, il livello di soddisfazione dei propri clienti.

I clienti chiedono ai propri fornitori prodotti e servizi in grado di soddisfare le loro esigenze di business, spinti dalla necessità di competere nel loro mercato. Le esigenze sono generalmente espresse in specifiche di prodotto o di servizio che poi vengono tradotte nei cosiddetti "requisiti del cliente". Tali requisiti sono generalmente specificati contrattualmente dal cliente, ma possono anche essere stabiliti dall'azienda stessa. In entrambi i casi, è sempre il cliente che esprime il proprio consenso all' "accettazione del prodotto finale". Inoltre, la continua evoluzione del mercato e delle tecnologie richiede la modifica dei requisiti iniziali e costringe le aziende software ad adeguare rapidamente i contenuti dei progetti in corso ed a migliorare di continuo i processi produttivi e gestionali.

L'approccio proposto dai sistemi qualità aderenti alle norme ISO 9000 suggeriscono alle organizzazioni di porre particolare cura nell'analizzare i requisiti del cliente, di mettere in atto processi adeguati per realizzare prodotti e servizi adeguati alle esigenze dei clienti ed a tenere sotto

controllo tali processi. Un buon sistema qualità costituisce la base per il miglioramento continuo delle prestazioni aziendali, focalizzandosi sulla soddisfazione dei clienti. Un sistema qualità così fatto garantisce ai clienti ed alle altre parti interessate la capacità dell'azienda a fornire prodotti e servizi che soddisfino i requisiti stabiliti.

2.1.5 Requisiti per il sistema qualità e per i prodotti

La famiglia di norme ISO 9000 distingue tra requisiti per i sistemi di gestione per la qualità e requisiti per i prodotti e servizi.

I requisiti per i sistemi di gestione per la qualità sono espressi nella norma ISO 9001. Essi sono di carattere generale, si applicano ad organizzazioni di qualsiasi dimensione e tipo e non esprimono alcun requisito specifico relativamente ai prodotti o servizi.

I requisiti per i prodotti possono specificati sia dai clienti, sia dalle organizzazioni stesse, come anticipazione dei requisiti del cliente, sia infine da disposizioni cogenti. I requisiti dei prodotti e dei servizi, ed in alcuni casi anche dei processi associati, sono generalmente descritti nelle "specifiche tecniche", nelle "norme di prodotto", in "accordi contrattuali" e nelle "normative cogenti".

2.1.6 Approccio dei sistemi qualità

L'approccio più efficace per la realizzazione di un sistema qualità è rappresentato dalle seguenti fasi:

- determinare le esigenze e le aspettative dei clienti e delle altre parti interessate;
- stabilire la politica e gli obiettivi per la qualità dell'azienda;
- determinare i processi e le responsabilità necessari per raggiungere gli obiettivi per la qualità;
- determinare e fornire le risorse necessarie per raggiungere gli obiettivi per la qualità;
- stabilire i metodi per misurare l'efficacia e l'efficienza dei processi definiti;
- mettere in atto queste misure per stabilire l'efficacia e l'efficienza raggiunta dai processi;
- determinare le modalità per evitare le non conformità ed eliminarne le cause;

- stabilire ed applicare un processo per il miglioramento continuo del sistema qualità.

Tale approccio può essere utilizzato anche per migliorare un sistema qualità già esistente.

Un'azienda che adotta un tale approccio acquista fiducia nei propri mezzi, garantisce la qualità dei propri prodotti e servizi, migliora la soddisfazione dei propri clienti, si dota di uno strumento efficace per il miglioramento continuo del proprio sistema qualità. In ultima analisi, assicura il successo delle proprie operazioni.

2.1.7 Approccio per processi

Un processo è un insieme di attività che utilizza risorse per trasformare elementi in ingresso in elementi in uscita.

Un'organizzazione deve identificare numerosi processi, interrelati ed integrati tra di loro, per gestire il complesso delle proprie operazioni produttive e gestionali. Un elemento in uscita da un processo è spesso l'elemento in entrata di un altro processo. L'identificazione e la gestione sistematica dei processi aziendali è chiamata "approccio per processi". La norma incoraggia l'adozione di un tale approccio.

La figura di sotto mostra il modello basato sui processi e l'importanza attribuita al cliente. Il modello rappresenta il modo per realizzare il miglioramento continuo del SGQ.

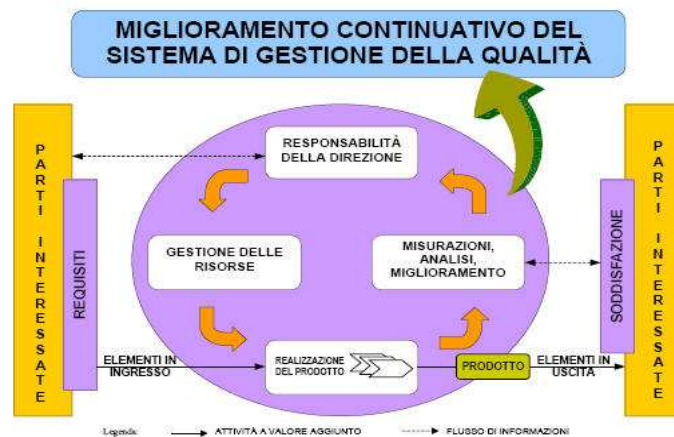


FIGURA 1 - Modello di sistema di gestione per la qualità basato sui processi.

Figura 1. Modello di sistema di gestione per la qualità basato sui processi.

2.1.8 **Politica ed obiettivi per la qualità**

La politica e gli obiettivi per la qualità servono a stabilire gli orientamenti dell'azienda e a guidare le azioni dell'organizzazione. Essi definiscono concreti risultati da raggiungere e guidano l'organizzazione nell'utilizzo delle risorse per raggiungerli.

La politica per la qualità definisce il quadro di riferimento entro il quale considerare (stabilire e riesaminare) gli obiettivi per la qualità.

Gli obiettivi, a loro volta, devono essere coerenti con la politica stabilita ed il miglioramento continuo perseguito, e devono essere misurabili.

I risultati conseguiti hanno un impatto positivo sulla qualità dei prodotti realizzati e dei servizi erogati. Inoltre, hanno un impatto positivo sull'efficacia delle operazioni produttive e sulle prestazioni economico-finanziarie. In ultima analisi, hanno un impatto positivo sul livello di soddisfazione dei clienti e di tutte le parti interessate.

2.1.9 **Ruolo dell'alta direzione**

La direzione aziendale (include tutto il management) deve esercitare la propria leadership per creare un ambiente che promuova, faciliti, e di fatto lo realizzi concretamente, in pieno coinvolgimento di tutto il personale. Il sistema qualità deve risultare efficace e pienamente compreso da tutti. I principi di gestione per la qualità, costituiscono la base su cui fonda il ruolo della direzione aziendale; questo consiste principalmente in:

- Stabilire la politica e gli obiettivi per la qualità dell'organizzazione e mantenerli aggiornati;
- Promuovere la politica e gli obiettivi per la qualità all'interno dell'intera organizzazione in modo che il personale si senta consapevole, coinvolto e motivato;
- Fare in modo che i requisiti dei clienti siano sempre indirizzati con la giusta attenzione;
- Assicurare che siano stabiliti processi appropriati che permettano di soddisfare i requisiti dei clienti e di raggiungere gli obiettivi per la qualità;
- Assicurare che sia predisposto, messo in atto e mantenuto aggiornato un sistema qualità, efficace ed efficiente, che permetta di raggiungere gli obiettivi per la qualità stabiliti;
- Assicurare la disponibilità delle risorse necessarie;

- Riesaminare periodicamente il sistema qualità;
- Prender decisioni relativamente alla politica ed agli obiettivi per la qualità;
- Intraprendere le azioni necessarie per il miglioramento del sistema qualità.

2.1.10 Documentazione del sistema qualità

a) Valore della documentazione

La documentazione permette di trasmettere le finalità dell'azienda all'organizzazione, ai clienti ed alle parti interessate. Essa permette inoltre di assicurare la coerenza delle azioni da intraprendere.

L'utilizzo della documentazione permette di:

- ottenere la conformità ai requisiti dei clienti ed a migliorare la qualità;
- fornire un addestramento adeguato;
- assicurare la reperibilità e la rintracciabilità;
- fornire evidenze oggettive;
- valutare l'efficacia e l'adeguatezza del sistema qualità.

La documentazione non sarà quindi mai prodotta fine a se stessa, ma costituire un'attività che aggiunge valore.

b) Tipi di documenti utilizzati nei sistemi qualità

I sistemi qualità utilizzano generalmente i seguenti tipi di documentazione:

- documenti che descrivono il sistema qualità e sono indirizzati sia all'interno che all'esterno dell'organizzazione; tali documenti sono detti "Manuali della qualità";
- documenti che descrivono come il sistema qualità si applica ad uno specifico prodotto, progetto o contratto; tali documenti sono detti "Piani della qualità";
- documenti che stabiliscono dei requisiti; tali documenti sono detti "Capitolati";
- documenti che forniscono raccomandazioni o consigli; tali documenti sono detti "Linee guida";

- documenti che forniscono informazioni su come svolgere coerentemente attività e processi; tali documenti sono detti "Procedure documentate", "Istruzioni operative";
- documenti che forniscono evidenza oggettiva di attività svolte o di risultati conseguiti; tali documenti sono detti "Registrazioni".

L'azienda stabilisce il numero, la tipologia e l'estensione della documentazione da produrre per il proprio sistema qualità, in base alle dimensioni dell'organizzazione, alle interrelazioni dei processi, alla complessità dei prodotti, dei requisiti, alla capacità del personale, ecc.

2.1.11 Valutazione dei sistemi di gestione per la qualità ...

a) Valutazione dei processi nell'ambito del sistema qualità

La valutazione del sistema qualità ha il compito di verificare, tra le altre cose, che "ciascun processo":

- sia identificato e definito in modo appropriato;
- siano state attribuite le responsabilità relative ad esso;
- siano attuate e tenute aggiornate le procedure stabilite;
- risulti efficace ai fini del conseguimento dei risultati attesi.

La valutazione del sistema qualità è quindi la risultante delle valutazioni di tutti i processi stabiliti. La verifica può essere fatta tramite verifiche ispettive e riesame del sistema qualità, autovalutazioni.

b) Verifiche ispettive sul sistema di gestione per la qualità

Le verifiche ispettive tendono a valutare il livello di aderenza del sistema qualità ai requisiti per esso stabiliti (norme ISO9000, standard e normative cogenti, ecc.). I risultati delle verifiche ispettive sono utilizzati per valutare l'efficacia del sistema qualità ed individuare le opportunità di miglioramento.

Le verifiche ispettive sono di tre tipi:

1. **Verifiche ispettive di prima parte**, sono condotte direttamente dalla stessa organizzazione o da altri per suo conto, e sono utilizzate per effettuare l'autovalutazione e/o per dichiarare la conformità del sistema qualità alle norme applicabili;
2. **Verifiche ispettive di seconda parte**, sono condotte dai clienti dell'azienda, direttamente o da altri per suo conto;

3. **Verifiche ispettive di terza parte**, sono condotte da organizzazioni esterne indipendenti, generalmente accreditate, e forniscono la certificazione rispetto ai requisiti applicabili (per esempio, ISO9001:2000).

La norma ISO 19011 costituisce una Linea guida per le verifiche ispettive.

c) Riesame del sistema di gestione per la qualità

La direzione aziendale, insieme a tutto il management, deve effettuare riesami periodici e sistematici del sistema qualità per valutare quanto esso sia appropriato, adeguato, efficace ed efficiente. La valutazione è fatta rispetto alla politica e agli obiettivi per la qualità. Il riesame può richiedere la necessità di adeguare la politica e gli obiettivi per la qualità alle mutate esigenze dei clienti e delle altre parti interessate. Il riesame comprende anche la determinazione delle necessità per opportune azioni.

Per effettuare i riesami si utilizzano tutte le informazioni disponibili e provenienti, per esempio, dai rapporti delle verifiche ispettive, dal risultato del rilevamento della soddisfazione dei clienti, ecc.

d) Autovalutazione

L'autovalutazione consiste in un riesame complessivo e sistematico delle attività svolte dall'organizzazione e dei risultati conseguiti, a fronte del sistema qualità o a fronte di un modello scelto come riferimento (per esempio, un modello di eccellenza come l'EFQM, un modello di maturità come il CMMI, ecc.).

I risultati dell'autovalutazione forniscono all'azienda un quadro complessivo delle prestazioni e del livello di maturità del proprio sistema qualità. Tale approccio può fornire anche utili indicazioni su possibili aree di intervento per il miglioramento del sistema.

2.1.12 Miglioramento continuo

Il miglioramento continuo del sistema qualità ha lo scopo principale di potenziare le capacità dell'azienda ad aumentare il livello di soddisfazione dei propri clienti ed a migliorare le prestazioni dell'organizzazione. Il miglioramento continuo è realizzato tramite le seguenti attività:

- Analisi e valutazione della situazione attuale con identificazione delle aree di intervento;
- Determinazione degli obiettivi di miglioramento da raggiungere;
- Ricerca delle possibili soluzioni (azioni) per raggiungere gli obiettivi stabiliti;
- Valutazione delle soluzioni individuate e determinazione di quelle da seguire;
- Attuazione delle azioni stabilite;
- Misurazione, verifica, analisi e valutazione dei risultati conseguiti a fronte di quelli attesi;
- Formalizzazione delle modifiche al sistema qualità sperimentate con successo.

I risultati sono quindi riesaminati per individuare nuove opportunità di miglioramento.

Al termine del ciclo mostrato sopra, quindi, si può avviare un nuovo ciclo con nuovi obiettivi.

E' facile individuare in tale approccio il classico ciclo di Deming: "Plan, Do, Check, Act".

L'approccio al miglioramento continuo diventa quindi uno strumento potente che permette all'azienda di accrescere progressivamente le proprie capacità e di mantenere la competitività adeguandosi al cambiamento continuo del mercato (clienti e concorrenti).

2.1.13 Ruolo delle tecniche statistiche

L'utilizzo di tecniche statistiche nell'analisi del proprio sistema qualità può aiutare l'organizzazione a comprendere la variabilità dei risultati ottenuti, a risolvere gli eventuali problemi che stanno all'origine dei prestazioni poco soddisfacenti ed a migliorare quindi l'efficacia e l'efficienza del sistema stesso.

Principalmente, tali tecniche aiutano ad individuare elementi di attenzione e a prendere decisioni opportune basate su dati di fatto.

La variabilità dei risultati può essere individuata a fronte di prestazioni complessive soddisfacenti. Ma individuare punti o momenti di flessione nell'andamento dei risultati può rappresentare un notevole aiuto per il miglioramento delle prestazioni stesse.

Tali variabilità possono essere riscontrate nelle caratteristiche misurate dei prodotti, servizi e processi, e possono verificarsi in tutte le fasi del ciclo di vita del software.

Le tecniche statistiche aiutano a misurare, descrivere, analizzare, interpretare e controllare le variabilità dei prodotti, servizi e processi.

Tali tecniche possono aiutare anche in presenza di pochi dati a disposizione fornendo indicazioni sugli andamenti delle misure ed individuando le aree da indagare con maggiore attenzione e profondità.

L'analisi statistica dei dati a disposizione permette di individuare le possibili cause di andamenti negativi e l'estensione dei fenomeni, aiutando a risolvere problemi esistenti ed a prevenirne di nuovi.

La norma ISO/TR 10017 fornisce una guida sull'uso delle tecniche statistiche nei sistemi di qualità

2.1.14 Scopi dei sistemi qualità e di altri sistemi di gestione

Il "sistema qualità" è parte integrante del sistema di gestione aziendale. Costituisce quella parte del sistema di gestione che fa riferimento agli obiettivi per la qualità e permette all'azienda di raggiungere risultati operativi in grado di soddisfare le esigenze, le aspettative ed i requisiti dei clienti e delle altre parti interessate. In altre parole, permette di aumentare la soddisfazione dei clienti nei confronti dell'azienda.

Gli obiettivi per la qualità sono complementari a quelli più operativi ed organizzativi dell'azienda quali, ad esempio, la redditività, la crescita, il finanziamento, l'ambiente, la salute e la sicurezza nei luoghi di lavoro, ecc.

Il sistema qualità può essere dunque integrato nel sistema generale di gestione aziendale in modo armonico e sinergico. Così possono essere utilizzati elementi comuni. L'individuazione di obiettivi comuni, la pianificazione delle risorse, la definizione dei processi, ecc. possono essere integrati armonicamente e costituire un unico quadro di maggiore potenzialità, efficacia ed efficienza.

Il sistema di gestione complessivo così realizzato può essere valutato e migliorato a fronte dei requisiti espressi dalle norme internazionali ISO 9001 e ISO 14001.

Le verifiche ispettive possono essere condotte sull'intero sistema di gestione aziendale oppure essere circoscritte al solo sistema qualità!

2.1.15 Relazione tra i sistemi qualità ed altri sistemi di gestione

La nuova versione delle norme ISO 9000 (Vision 2000), come già detto, si è evoluta notevolmente. Essa offre oggi uno schema completo per il miglioramento delle prestazioni aziendali. La nuova struttura delle norme ISO 9000 si è uniformata ai "modelli di eccellenza" esistenti.

Con i modelli di eccellenza la norma condivide i principi generali e permette all'organizzazione di:

- Individuare i propri punti di forza e quelli di debolezza;
- Valutare il proprio sistema a fronte di modelli stabiliti;
- Utilizzare il modello come base per il miglioramento continuo;
- Ottenere riconoscimenti esterni riconosciuti.

Una possibile differenza tra l'approccio proposto dalla norma ISO 9000 e quello proposto dai modelli di eccellenza può essere nelle finalità dei vari approcci.

La norma ISO 9000 definisce i requisiti per un sistema qualità e la valutazione è fatta a fronte del livello di aderenza a tali requisiti.

I modelli di eccellenza tendono a confrontare il sistema di un'organizzazione con sistemi di altre aziende.

2.1.16 Termini e definizioni

La norma riporta le definizioni dei termini utilizzati, raggruppati nelle seguenti tipologie:

- Termini relativi alla **qualità** (qualità, requisito, classe, soddisfazione del cliente, capacità);
- Termini relativi alla **gestione** (sistema, sistema di gestione, sistema di gestione per la qualità, politica per la qualità, obiettivo per la qualità, gestione, gestione per la qualità, alta direzione, vertice, pianificazione della qualità, controllo della qualità, assicurazione della qualità, miglioramento della qualità, miglioramento continuo, efficacia, efficienza);
- Termini relativi all'**organizzazione** (organizzazione, struttura organizzativa, infrastruttura, ambiente di lavoro, cliente, fornitore, parte interessata);
- Termini relativi al **processo** e al **prodotto** (processo, prodotto, progetto, progettazione e sviluppo, procedura);

- Termini relativi alle **caratteristiche** (caratteristica, caratteristica qualitativa, fidatezza, rintracciabilità);
- Termini relativi alla **conformità** (conformità, non conformità, difetto, azione preventiva, azione correttiva, correzione, rilavorazione, riclassificazione, riparazione, scarto, concessione, deroga, rilascio);
- Termini relativi alla **documentazione** (informazione, documento, specifica, manuale della qualità, piano della qualità, registrazione);
- Termini relativi all'**esame** (evidenza oggettiva, ispezione, prova, verifica, validazione, processo di qualifica, riesame);
- Termini relativi alla **verifica ispettiva** (verifica ispettiva/audit, programma delle verifiche ispettive, criteri della verifica ispettiva, evidenza della verifica ispettiva, risultanza della verifica ispettiva, conclusione della verifica ispettiva, committente della verifica ispettiva, organizzazione verificata, valutatore, gruppo di verifica ispettiva, esperto tecnico, competenza);
- Termini relativi all'**assicurazione della qualità** per i processi di misurazione (sistema di controllo della misurazione, processo di misurazione, conferma metrologica, apparecchiatura per misurazione, caratteristica metrologica, funzione ,metrologica);

2.2 ISO 9001:2000, Sistemi di gestione per la qualità - Requisiti

2.2.1 Contenuti della norma

"La norma specifica i requisiti di un modello di sistema di gestione per la qualità per tutte le organizzazioni, indipendentemente dal tipo e dimensione delle stesse e dai prodotti forniti. Essa può essere utilizzata per uso interno, per scopi contrattuali e di certificazione." (definizione tratta dalla norma stessa).

La norma descrive quanto richiesto in otto capitoli:

1. SCOPO E CAMPO DI APPLICAZIONE
2. RIFERIMENTI NORMATIVI
3. TERMINI E DEFINIZIONI
4. SISTEMA DI GESTIONE PER LA QUALITA'
5. RESPONSABILITA' DELLA DIREZIONE

6. GESTIONE DELLE RISORSE

7. REALIZZAZIONE DEL PRODOTTO

8. MISURAZIONI, ANALISI E MIGLIORAMENTO

Dettagli sui contenuti di ciascun capitolo e relativi paragrafi e sottoparagrafi sono forniti nello schema del "Manuale della qualità" proposto.

2.2.2 Relazione con la ISO 9004

La versione attuale (Vision 2000) delle norme ISO 9001 e ISO 9004 sono state realizzate in modo coerente e complementare. Quindi esse vanno utilizzate "in coppia".

La ISO 9001 definisce i requisiti per un sistema qualità da utilizzare sia in ambito interno che per fini contrattuali o per la certificazione. Essa è focalizzata sull'efficacia del sistema qualità e la sua capacità di soddisfare i requisiti dei clienti.

La ISO 9004 fornisce l'orientamento per il miglioramento continuo del sistema qualità. Essa rappresenta una guida per l'organizzazione che voglia migliorare le proprie prestazioni in termini di efficacia ed efficienza.

2.3 ISO 9004, Guida al miglioramento continuo

2.3.1 Contenuti della norma

"La norma fornisce delle linee guida complementari ai requisiti della ISO 9001 per migliorare l'efficacia e l'efficienza di un sistema di gestione per la qualità e le prestazioni dell'organizzazione. Essa è applicabile ai processi di un'organizzazione indipendentemente dal tipo e dimensioni della stessa e dai prodotti forniti. Scopo principale della norma è il conseguimento di un miglioramento progressivo attraverso la soddisfazione dei clienti e delle parti interessate (dipendenti, collettività, soci dell'organizzazione, fornitori)." (definizione tratta dalla norma stessa).

I primi tre capitoli descrivono la norma in termini di scopo e campo di applicazione, riferimenti normativi e termini e definizioni applicabili.

I successivi cinque capitoli descrivono le linee guida per migliorare il sistema qualità secondo i quattro componenti fondamentali: 1. Ruolo e responsabilità della direzione nei confronti del sistema qualità; 2. Gestione delle risorse umane e tecnologiche; 3. Processi produttivi e

gestionali per la realizzazione dei prodotti e servizi forniti e per il governo dell'organizzazione; 4. Misurazioni, analisi e miglioramento del sistema qualità.

Data la simmetria, coerenza e complementarietà della norma ISO 9004 con la ISO 9001, i contenuti della presente norma sono presentati con la stessa sequenza e terminologia come mostrato dall'indice riportato qui di seguito.

1. SCOPO E CAMPO DI APPLICAZIONE
2. RIFERIMENTI NORMATIVI
3. TERMINI E DEFINIZIONI
4. SISTEMA DI GESTIONE PER LA QUALITA'
5. RESPONSABILITA' DELLA DIREZIONE
6. GESTIONE DELLE RISORSE
7. REALIZZAZIONE DEL PRODOTTO
8. MISURAZIONI, ANALISI E MIGLIORAMENTO

2.4 ISO 19011, Guida alle verifiche ispettive

2.4.1 Contenuti della norma

La norma descrive le Linee guida per le valutazioni (audit) dei sistemi di gestione per la qualità e dei sistemi di gestione ambientale.

La versione attuale della norma (2003 nella versione in italiano) unifica le regole da seguire nelle verifiche ispettive (audit) sui sistemi di gestione (per la qualità ed ambientale).

La norma fornisce delle linee guida sui seguenti elementi:

- Principi dell'attività di audit (solo per quanto concerne la certificazione ISO 9000),
- Gestione dei programmi di audit,
- Conduzione dell'audit del sistema di gestione per la qualità e del sistema di gestione ambientale,
- Competenza degli auditor di tali sistemi di gestione.

La è destinata ad una platea molto ampia di utilizzatori: gli auditor in primo luogo; le organizzazioni che realizzano il proprio sistema di gestione per la qualità e/o di gestione ambientale; le organizzazioni che operano nella certificazione o nella formazione ed addestramento degli auditor; quelle che operano nella certificazione di sistemi di gestione, nell'accreditamento o nella normazione nel campo della valutazione della conformità.

2.5 ISO/IEC 9126, Modello per la qualità del software

2.5.1 Contenuti della norma

La qualità dei prodotti software è definita nelle norme della serie ISO/IEC 9126.

Tali norme definiscono il modello e le relative caratteristiche del software.

La prima (ISO/IEC 9126-1) definisce il modello per la qualità dei prodotti software.

Le altre tre (ISO/IEC TR 9126-2, 3, 4) partono dal modello appena menzionato e definiscono le caratteristiche e le relative metriche per la qualità "esterna", "interna" ed "in uso".

Le caratteristiche della qualità dei prodotti software sono definite come: Funzionalità, Affidabilità, Usabilità, Efficienza, Manutenibilità, Portabilità.

Per ogni caratteristica sono definite sotto-caratteristiche (attributi) che identificano i diversi componenti del software.

L'insieme di tutte le caratteristiche ed i relativi attributi costituiscono il modello per la qualità interna ed esterna dei prodotti software.

Nel dettaglio, le norme in oggetto sono:

- ISO/IEC 9126-1:2001, Software engineering - Product Quality Part 1: Quality Model;
- ISO/IEC 9126-2:2002, Software engineering - Product Quality Part 2: External metrics;
- ISO/IEC 9126-3:2002, Software engineering - Product Quality Part 3: Internal metrics;

- ISO/IEC 9126-4:2002, Software engineering - Product Quality Part 4: Quality in use.

Una descrizione delle norme e del loro utilizzo da parte delle organizzazioni software è fornita dall'autore in un documento disponibile in questo stesso sito:

Ercole F. Colonese, La qualità del software secondo il modello ISO/IEC 9126 - Modello, metriche e modalità di utilizzo della qualità interna, esterna ed in uso nei progetti reali, Versione 2.0 - Giugno 2006.

2.6 ISO 12207, Norme sul ciclo di vita del software

2.6.1 Una famiglia di norme per la gestione del ciclo di vita

Queste norme indirizzano i processi di produzione ed il ciclo di vita del software. Esse definiscono un modello base per i processi del ciclo di vita del software secondo l'approccio più moderno. Esse sono norme generali e devono essere personalizzate alle esigenze della propria organizzazione. Sono fornite linee guida per il loro utilizzo nei progetti di sviluppo software.

- ISO/IEC 12207:1995, Information Technology - Software Life Cycle Processes;
- ISO/IEC 12207:1995/Amd.1:2002, Information Technology - Software Life Cycle Processes - Amendment 1;
- ISO/IEC TR 15271: 1998, Information Technology - Guide for ISO/IEC 12207 (Software Life cycle Process);
- ISO/IEC TR 16326:1999, Software Engineering - Guide for the Application of ISO/IEC 12207 to Project Management.

2.6.2 Principali differenze con la ISO 9001

In sintesi le principali differenze fra ISO 9001 e ISO 12207 sono quelle mostrate nella tabella che segue.

Maggiori dettagli sui contenuti delle norme sono disponibili selezionando le rispettive norme.

Tabella 1. Differenze tra le norme ISO 9001 e ISO 12207.

ISO 9001	ISO 12207
• Si applica ad organizzazioni di qualsiasi genere, dimensione e tipo di prodotti forniti; • Si occupa della qualità dal punto di vista dell'azienda; • Definisce specifiche attività di gestione; • E' utilizzata per la certificazione del sistema qualità del fornitore.	• Si applica ad organizzazioni di produzione e di servizio nel campo del software; • Si concentra sul ciclo di vita del software; • Si occupa di tutte le attività del ciclo di vita del software, sia gestionali che tecniche; • E' utilizzata per acquisire, fornire, sviluppare, utilizzare e mantenere un prodotto software, e non prevede procedure per la valutazione del fornitore.

2.7 ISO 12207, Norme sul ciclo di vita del software

2.7.1 Contenuti della norma

La norma ha lo scopo principale di definire in modo preciso e condiviso i processi del Ciclo di Vita del Software: dalla formulazione dei requisiti, allo sviluppo, all'esercizio ed alla manutenzione.

La norma emessa nel 1995 è stata aggiornata nel 2002.

I processi produttivi e gestionali delle organizzazioni sono suddivisi dalla norma in tre categorie:

1. **primari**, comprendenti le attività direttamente legate allo sviluppo del software,
2. **di supporto**, che includono la gestione dei documenti e dei processi di controllo della qualità,
3. **organizzativi**, che coprono gli aspetti manageriali e di gestione delle risorse.

Nota: selezionare le singole categorie per avere maggiori dettagli.

Per ciascun processo la norma evidenzia chiaramente:

- obiettivo e responsabilità,
- lista delle attività che lo compongono,
- singoli compiti nei quali è suddivisa ogni attività.

La norma fa proprio il ciclo di miglioramento dei processi basato sulla sequenza Plan-Do-Check-Act (pianifica, esegui, controlla, attua eventuali modifiche).

L'Aicq-ci ha realizzato un apposito quaderno che tratta l'argomento e fornisce validi suggerimenti su come applicare la norma ISO 12207 nelle organizzazioni software.

"Quaderno N. 17. Settembre 2004. Modello dei processi ISO 9000 applicato in aziende di sviluppo software e fornitura di servizi IT. Aicq-ci"

2.7.2 Processi primari

I processi primari definiti dalla norma ISO 12207 indirizzano le seguenti attività principali:

- Acquisizione,
- Fornitura,
- Sviluppo,
- Esercizio,
- Manutenzione.

Ciascun processo primario è definito e descritto dalla norma in termini di attività (activities) e compiti (tasks).

Ogni compito (task), a sua volta, indica cosa occorre fare ("what to do") e non "come" la si deve fare ("how to do").

In particolare, la norma fa uso dei seguenti "verbi" per indicare i diversi livelli di obbligatorietà:

- dovrà, per la definizione dei requisiti;
- dovrebbe per specificare le raccomandazioni;
- può, per indicare il permesso a fare qualcosa, o il fatto di essere in grado di farla;
- qualunque altro (tempo presente), per descrivere un preambolo o per descrivere il contesto.

Per una descrizione completa e dettagliata della norma occorre fare riferimento alla norma stessa.

2.7.3 Processi di supporto

I processi di supporto definiti dalla norma ISO 12207 indirizzano le seguenti attività principali:

- Gestione della documentazione;
- Gestione della configurazione;
- Assicurazione della qualità
- Verifica;
- Validazione;
- Revisioni congiunte;
- Audit;
- Risoluzione dei problemi.

I processi di supporto aiutano le attività di tutti gli altri processi dell'organizzazione a garantire il successo e la qualità del progetto.

Questi processi possono essere attivati da un processo primario o da un altro processo di supporto.

2.7.4 Processi organizzativi

I processi organizzativi definiti dalla norma ISO 12207 indirizzano le seguenti attività di gestionali:

- Gestione dello sviluppo,
- Gestione delle infrastrutture,
- Gestione del miglioramento,
- Formazione.

Tali processi eseguono funzioni a livello organizzativo aziendale, per supportare altri processi primari, di supporto o organizzativi.

I processi organizzativi aiutano nel definire, controllare e migliorare gli altri processi.

Per una descrizione completa e dettagliata della norma occorre fare riferimento alla norma stessa.

2.8 ISO/IEC TR 15271, Guida all'utilizzo della norma ISO/IEC 12207 sul ciclo di vita del software

2.8.1 Contenuti della norma

La norma ISO/IEC TR 15271 è un rapporto tecnico (TR) emesso nel 1998 con lo scopo di fornire una guida all'applicazione della norma ISO/IEC 12207 relativamente al ciclo di vita del software.

Il rapporto tecnico tiene conto dei vari fattori da considerare quando si applica la norma nei diversi cicli di vita del software delle organizzazioni.

La guida non fornisce quindi nuovi requisiti da applicare ai cicli di vita ma piuttosto consiglia come interpretare ed applicare quelli (i requisiti) già espressi dalla norma ISO/IEC 12207.

La guida discute tre modelli fondamentali di cicli di vita e fornisce esempi di come personalizzarli nelle diverse realtà.

Maggiori dettagli sui contenuti delle norme sono disponibili selezionando le rispettive norme.

2.9 ISO/IEC TR 16326, Guida all'utilizzo della norma ISO/IEC 12207 nella gestione dei progetti software

2.9.1 Contenuti della norma

La norma ISO/IEC TR 16326 è un rapporto tecnico (TR) emesso nel 1998 con lo scopo di fornire una guida all'applicazione della norma ISO 12207 nella gestione dei progetti (project management).

Il rapporto tecnico indirizza gli aspetti della gestione dei progetti specifici per lo sviluppo o la manutenzione del software.

In particolare, fornisce una guida per la gestione dei problemi che tradizionalmente affliggono i progetti software: requisiti omessi o non indirizzati correttamente, ritardo nei tempi di consegna, costi superiori a quelli pianificati, ecc.

Il rapporto tecnico presenta i processi del ciclo di vita del software secondo la prospettiva del Project Management. Fornisce suggerimenti utili, basati sull'esperienza, su quanto imparato dalla pratica, ecc. Presenta le migliori pratiche disponibili (best practices) al riguardo. Fornisce inoltre raccomandazioni su come gestire le varie attività di un progetto da parte dei singoli addetti (practitioners).

Infine, il rapporto tecnico mostra ai vari addetti ai lavori (capo progetto, tecnici, staff, ecc.) come integrare le proprie attività con quelle degli altri nell'ambito del ciclo di vita.

Maggiori dettagli sui contenuti delle norme sono disponibili selezionando le rispettive norme.

3. Il modello CMMI nelle organizzazioni software

Il modello CMMI® (Capability Maturity Model®, Integration), messo a punto dal SEI® (Software Engineering Institute) fornisce un modello basato sul livello di maturità dei singoli processi di sviluppo.

Il grado di maturità dei processi è definito su di una scala a cinque livelli. A ciascun livello di maturità sono associate aree di competenza (KPA - Key Process Area).

Il modello ha due rappresentazioni: "staged" e "continuous".

In continua evoluzione, il modello è stato di recente presentato nella sua versione 1.2 che presenta alcune importanti modifiche rispetto alla versione precedente.

Livelli di maturità

Il grado di maturità dei processi è definito su di una scala a cinque livelli che, in ordine di maturità crescente, sono:

1. Initial
2. Managed
3. Defined
4. Quantitatively Managed
5. Optimizing

I livelli di maturità consistono di un insieme di predefinito di aree di processo. I livelli di maturità sono misurati dal raggiungimento degli obiettivi specifici e generici definiti per ciascuna area di processo.

La descrizione dei cinque livelli di maturità è fornita nella pagina di questo sito che descrive il modello.

Aree di processo (KPA)

Per ciascun livello di maturità è definito un insieme di "pratiche" che permettono, quando eseguire correttamente, di raggiungere gli obiettivi definiti e considerati importanti per l'area stessa. Le pratiche sono azioni da eseguire per raggiungere gli obiettivi.

Il modello CMMI definisce le KPA come aree di processo focalizzate su specifici ambiti.

3.1 Livelli di maturità

I cinque livelli di maturità del modello sono rappresentati nella figura che segue.

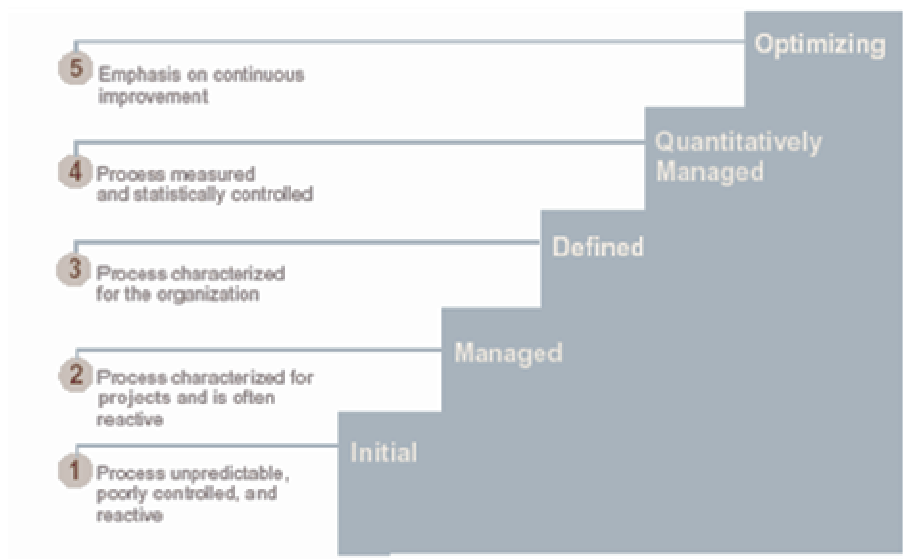


Figura 2. I cinque livelli di maturità del modello CMM.

I livelli di maturità consistono di un insieme di predefinito di aree di processo. I livelli di maturità sono misurati dal raggiungimento degli obiettivi specifici e generici definiti per ciascuna area di processo.

Nota: i nomi dei cinque livelli di maturità riportati nella figura si riferiscono alla rappresentazione "staged" del modello. Nella rappresentazione "continuous" del modello esiste anche un livello inferiore (Livello zero) denominato "Incomplete". Sempre nella versione continuous, il primo livello di maturità è denominato "Performed" e non "Initial" come nella versione staged.

3.1.1 Livello di maturità 1: Initial

Al livello di maturità iniziale, i processi sono generalmente ad hoc e caotici. L'organizzazione abitualmente non dispone di ambienti di sviluppo stabili. Il successo raggiunto da tali organizzazioni dipende

esclusivamente dalla competenza e dall'eroismo delle persone e non sull'utilizzo di processi consolidati. A dispetto di tali ambienti caotici e ad hoc, tali organizzazioni possono anche realizzare buoni prodotti e servizi software; tuttavia essi non possono prevedere i risultati finali e spesso non riescono a rispettare i tempi di consegna ed il budget.

Le organizzazioni che sono al livello di maturità 1 tendono a sottovalutare gli impegni e ad assumersene più di quanti ne possano rispettare, sono indotte ad abbandonare i processi nei momenti di crisi e non sono in grado di ripetere i successi ottenuti in precedenza.

3.1.2 Livello di maturità 2: Managed

Al livello di maturità 2, un'organizzazione ha raggiunto tutti gli obiettivi specifici e quelli generici definiti per le aree di processo associate al livello. In altre parole, i progetti sviluppati dall'organizzazione assicurano che i requisiti sono gestiti e che i processi sono pianificati, eseguiti, misurati e controllati.

Il livello di maturità raggiunto dall'organizzazione assicura che i processi sono seguiti anche in condizione di stress. Le pratiche assicurano che i progetti sono gestiti in accordo con i piani documentati. I requisiti, i processi, i prodotti intermedi realizzati, i servizi sono tutti gestiti. Lo stato degli artefatti realizzati e dei prodotti rilasciati e dei servizi erogati sono visibili al management come atteso.

Gli impegni sono stabiliti tenendo conto di tutti gli attori coinvolti (stakeholders) e da essi rivisti. Questi rivedono e controllano gli artefatti ed i servizi per verificare che soddisfino i requisiti, gli standard e gli obiettivi.

3.1.3 Livello di maturità 3: Defined

Al livello di maturità 3, un'organizzazione ha raggiunto tutti gli obiettivi definiti per il livello di maturità precedente (livello 2) più tutti quelli definiti per le aree del presente livello. A questo livello di maturità tutti i processi sono ben caratterizzati e compresi dall'organizzazione, definiscono gli standard, le procedure, gli strumenti ed i metodi.

L'insieme dei processi definiti costituiscono uno standard per l'organizzazione. Essi (i processi) sono utilizzati da tutti e migliorati con regolarità nella loro efficacia ed efficienza. I progetti stabiliscono quali processi utilizzare e li personalizzano per adattarli alle loro esigenze secondo regole stabilite.

Il management dell'organizzazione stabilisce gli obiettivi dei processi in base agli obiettivi dell'organizzazione stessa e ne assicura il controllo.

La caratteristica più saliente del livello di maturità 3 è quella di stabilire processi comuni per l'intera organizzazione e linee guida per la loro personalizzazione nei diversi progetti.

3.1.4 Livello di maturità 4: Quantitatively Managed

Al livello di maturità 4, un'organizzazione ha raggiunto tutti gli obiettivi definiti per i precedenti livelli (livello 2 e 3) più tutti quelli definiti per il livello attuale. Sono definiti i sottoprocessi che maggiormente contribuiscono alle performance generali dei processi stessi. I sottoprocessi selezionati sono controllati utilizzando tecniche statistiche o altre tecniche quantitative.

Gli obiettivi quantitativi stabiliti per valutare la qualità e le performance dei processi costituiscono i criteri utilizzati per la gestione dei processi stessi. Tali obiettivi indirizzano le necessità del cliente, degli utenti finali, dell'organizzazione e dei responsabili della gestione dei processi.

I processi sono misurati e valutati nelle loro performance tramite la raccolta e l'analisi statistica dei dati delle misurazioni effettuate sui risultati ottenuti. Eventuali variazioni dai valori attesi sono analizzate e le cause rimosse con opportune azioni correttive che ne prevengano il ripetersi.

Le misure sulla qualità e le performance dei processi sono memorizzate nell'apposito archivio in modo da supportare con i fatti il processo decisionale dell'organizzazione.

La caratteristica più saliente del livello di maturità 4 è quello di permettere una previsione dei risultati dei processi con sufficiente accuratezza a seguito del controllo statistico effettuato.

3.1.5 Livello di maturità 5: Optimizing

Al livello di maturità 5, un'organizzazione ha raggiunto tutti gli obiettivi definiti per i precedenti livelli (livello 2, 3 e 4) più tutti quelli definiti per il livello attuale. I processi sono migliorati di continuo in base alla comprensione quantitativa delle "cause comuni" delle varianze statistiche sui risultati dei processi.

Il miglioramento continuo delle performance dei processi è raggiunto tramite azioni incrementalì ed innovative delle tecnologie utilizzate. Gli obiettivi di miglioramento dei processi sono stabiliti in base agli

obiettivi di business dell'organizzazione. Essi sono quindi definiti, misurati e revisionati di continuo per riflettere eventuali cambiamenti negli obiettivi di business dell'organizzazione.

La caratteristica più saliente del livello di maturità 5 consiste nel tipo di varianza statistica indirizzata dalle azioni di miglioramento dei processi. Nel livello di maturità 4 sono indirizzate le "cause speciali" che producono varianze sui processi e ne impediscono la predicibilità statistica dei risultati. Nel livello di maturità 5 sono indirizzate le "cause comuni" e sono modificati i processi stessi per migliorarne le performance e mantenere la predicibilità dei risultati.

3.2 Aree di processo (KPAs)

Il grado di maturità dei processi è definito su di una scala a cinque livelli.

A ciascun livello di maturità sono associate aree di competenza (KPA - Key Process Area).

In continua evoluzione, il modello è stato di recente presentato nella sua versione 1.2 che presenta alcune importanti modifiche rispetto alla versione precedente.

Per ciascun livello di maturità è definito un insieme di "pratiche" che permettono, quando eseguire correttamente, di raggiungere gli obiettivi definiti e considerati importanti per l'area stessa. Le pratiche sono azioni da eseguire per raggiungere gli obiettivi.

Il modello CMMI definisce le seguenti aree di processo (KPA) focalizzate su specifici ambiti.

Tabella 2. I processi associati ai cinque livelli di maturità del modello CMM.

Livello	Focus	Area di processo
5. Optimizing	Continuous Process Improvement	Organizational Innovation and Deployment (OID) Causal Analysis and Resolution (CAR)
4. Managed	Quantitative Management	Organizational Process Performance (OPP) Quantitative Project Management (QPM)
3. Defined	Process Standardisation	Requirements Development (RD) Technical Solution (TS) Product Integration (PI) Organizational Environment for Integration (OEI) Validation (VAL) Verification (VER) Organizational Process Focus (OPF) Organizational Process Definition (OPD) Organizational Training (OT) Integrated Project Management (IPM) Risk Management (RSKM) Integrated Teaming (IT) Decision Analysis and Resolution (DAR)
2. Repeatable	Basic Project Management	Requirements Management (RM) Project Planning (PP) Project Monitoring and Control (PMC) Supplier Agreement Management (SAM) Measurement and Analysis (MEA) Process and Product Quality Assurance (PPQA) Configuration Management (CM)
1. Initial		Nessuna

4. La professione dell'ingegnere del software

A dispetto dei milioni di professionisti del software che operano a livello mondiale e della vastità del software presente nella società moderna, l'ingegneria del software non ha raggiunto ancora lo stato di legittima disciplina e di professione riconosciuta.

Solo dal 1993 un'apposita organizzazione, il Software Engineering Coordinating Committee (SWECC) opera attivamente per promuovere la professione del Software Engineer (Ingegnere del software) e della relativa disciplina del Software Engineering (Ingegneria del software). Il comitato è stato fondato dall'IEEE Computer Society e dall'Association for Computing Machinery (ACM).

Un primo traguardo nel definire una disciplina è quello di raggiungere un consenso sui contenuti di base della professione (Core Body of Knowledge). Operando in questa direzione, il comitato ha definito il corpo delle competenze di base della nuova professione.

E' stato pubblicato un documento che fornisce una descrizione di massima di tali competenze (SWEBOK - Software Engineering Body Of Knowledge).

4.1 Che cos'è la Software Engineering?

La IEEE Computer Society definisce l'ingegneria del software come:

“(1) L'applicazione di un sistematico, disciplinato e quantificabile approccio allo sviluppo, all'operatività e alla manutenzione del software; cioè l'applicazione dell'ingegneria al software.

(2) Lo studio degli approcci definiti al punto precedente.”¹

¹ “IEEE Standard Glossary of Software Engineering Terminology, 1990.”

4.2 Che cos'è una professione riconosciuta?

Affinché l'ingegneria del software sia legittimata come disciplina dell'ingegneria e sia riconosciuta come professione occorre raggiungere il consenso sulle competenze di base (Core Body Of Knowledge).

4.3 Quali sono le caratteristiche di una professione?

Le caratteristiche principali della professione ingegneristica includono le seguenti componenti².

- Formazione iniziale eseguita all'interno di un curriculum validato da un'organizzazione apposita tramite l'accreditamento (es.: frequenza dei corsi di un piano di formazione definito e superamento degli esami previsti);
- Registrazione dell'aderenza alle pratiche riconosciute tramite il superamento di esami particolari (esempio: iscrizione all'albo degli ingegneri, certificazione interna, ecc.);
- Sviluppo continuo delle competenze tramite frequenza di corsi di specializzazione;
- Supporto da parte di una comunità accreditata;
- Impegno a seguire un comportamento etico.

L'insieme degli articoli che ci si appresta a scrivere e a divulgare intendono coprire i primi tre punti.

4.4 Le competenze dell'ingegnere del software (SWEBOK)

4.4.1 Le 10 aree di competenza (Knowledge Areas)

Di seguito è fornita la lista delle 10 aree di competenza previste dal modello SWEBOK. Le prime cinque sono riportate nell'ordine in cui generalmente sono svolte nel ciclo di sviluppo (a cascata). Le rimanenti cinque sono generalmente svolte lungo l'intero ciclo di sviluppo e sono

² G. Ford and E. Gibbs, "A Mature Profession of Software Engineering" Software Engineering Institute, Carnegie Mellon University, Pittsburg, Pennsylvania, January 1996.

riportate in ordine alfabetico. Ciascuna area di competenza è composta da più sotto-aree che indirizzano argomenti specifici.

1. Software Requirements;
2. Software Design;
3. Software Construction;
4. Software Testing;
5. Software Maintenance;
6. Software Configuration Management;
7. Software Engineering Management;
8. Software Engineering Process;
9. Software Engineering Tools and Methods;
10. Software Quality.

Alcuni dettagli sulle singole aree di competenza sono forniti nel documento disponibile in questo stesso sito:

"Ercole F. Colonese, Introduzione al Software Engineering Body Of Knowledge (SWEBOK), 2004".

L'argomento può essere approfondito direttamente nel sito ufficiale SWEBOK.

5. Una metodologia a supporto dello sviluppo software

Lo sviluppo del software, come si è detto, è un'attività creativa ad alto contenuto umano e quindi soggetta ad errori. La presenza di errori nel software causa malfunzionamenti che dovranno essere corretti; e questo ha un costo. Il costo degli interventi sul software dipende dalla sua complessità e dal momento in cui gli errori sono rilevati: più complesso è il codice e più costoso sarà modificarlo, più tardi sono scoperti gli errori e più costoso risulterà risolverli! Ma il codice è regolarmente modificato anche dopo il suo rilascio in esercizio per potenziarne le funzionalità, per adeguarlo a normative o standard, per migliorarne le performance o per altri motivi ancora.

Come affrontare allora la problematica in modo strutturato ed ... ingegneristico?

L'Ingegneria del software è la disciplina che studia le problematiche relative a tale attività e fornisce soluzioni per affrontarle in modo sistematico. I metodi e le tecniche proposti negli anni e contraddistinti dalla qualità dei risultati ottenuti costituiscono un patrimonio di grande valore e disponibili come "best practices".

Il processo maturo ...

Il processo è l'elemento cui si è dato tradizionalmente forse maggiore importanza.

Il modello CMMI® (Capability Maturity Model®, Integration), messo a punto dal SEI® (Software Engineering Institute) fornisce un modello basato sul livello di maturità dei singoli processi di sviluppo. Il grado di maturità dei processi è definito su di una scala a cinque livelli. A ciascun livello di maturità sono associate aree di competenza (KPA - Key Process Area).

Ma adoperare un processo maturo non basta per sviluppare software di qualità. La competenza delle persone coinvolte nelle diverse attività è fondamentale: sviluppare software, si è detto, è un'attività creativa ad "alto contenuto umano".

Le competenze richieste ...

Le competenze si acquisiscono con un giusto mix di formazione teorica ed esperienza pratica sul campo. Ma la formazione è spesso trascurata

restringendola solo ai temi tecnologici (linguaggi, piattaforme, sottosistemi ecc.) e indirizzando poco (o quasi per niente) i metodi e le tecniche. Diversamente da quanto stabilito per altre professioni, per il software non sono definite regole e comportamenti. Ancora oggi non è riconosciuta legalmente la professione di Ingegnere del software: quindi non sono definite competenze e comportamenti, diritti e doveri. Negli USA sono nate iniziative con lo scopo di definire la professione e creare un albo. Alla professione sono richieste competenze specifiche, ruoli comportamentali ed etici e riconoscimenti legali.

Alla professione di Ingegnere del software è richiesta una competenza di base su temi specifici, identificati come "Software Engineering Body of Knowledge" (SWEBOK), da acquisire presso università o scuole di specializzazione.

I metodi, le tecniche, gli strumenti e le metriche a supporto

...

Alla "competenza delle persone" ed al "processo maturo" occorre aggiungere un terzo elemento fondamentale per lo sviluppo del software: l'utilizzo di adeguati metodi, tecniche, strumenti e metriche!

Così come un buon piatto dipende dalla bravura del cuoco, dall'originalità della ricetta e dalla genuinità degli ingredienti, così un software di qualità dipende dalle competenze di chi lo sviluppa, dalla maturità del processo seguito e dall'adeguatezza degli strumenti adoperati!

La metodologia proposta risulta semplice ed efficace, completa e rigorosa, facilmente adottabile da ogni organizzazione di sviluppo software. Sperimentata con successo da molte piccole e medie imprese di software, la metodologia adotta il modello CMMI, è aderente alle norme ISO9001:2000 e utilizza le migliori pratiche disponibili ("best practices").

La metodologia definita poggia su tre pilastri:

1. competenze delle persone;
2. processi maturi per lo sviluppo software;
3. metodi, tecniche, strumenti e metriche a supporto.

5.1 Competenze delle persone ...

La competenza delle persone coinvolte in un progetto software costituisce il pilastro portante di tutta l'impalcatura metodologica.

Essa si acquisisce con un giusto mix di formazione teorica ed esperienza pratica sul campo.

Ma la formazione è spesso trascurata o ristretta ai soli temi tecnologici (linguaggi di programmazione, piattaforme, sistemi ecc.). Poco (o quasi niente) è invece dedicato a metodi e tecniche.

5.1.1 Ruoli, responsabilità e competenze necessarie

La metodologia proposta indica i ruoli coinvolti nello sviluppo del software, le responsabilità ad esso associate e quali competenze debbano essere presenti per garantire la piena efficacia del ruolo stabilito. Gli elementi da prendere in considerazione sono quindi:

- i ruoli coinvolti nello sviluppo del software;
- le responsabilità associate a ciascun ruolo;
- le competenze necessarie perchè il ruolo possa essere ricoperto con la giusta efficacia.

Lo sviluppo del software è un'attività critica che richiede competenze adeguate a tutti i livelli dell'organizzazione: dall'alta direzione alle funzioni commerciali, dalla gestione dei progetti all'analisi e progettazione, dalla codifica ai test, dalla manutenzione ai ruoli di supporto come la gestione della configurazione e degli ambienti, l'assicurazione della qualità, la formazione.

5.1.2 SWEBOK ...

Diversamente da quanto stabilito per altri settori dell'ingegneria, per il software non sono definite formalmente le competenze necessarie per ricoprire i vari ruoli previsti. Ancora oggi la professione di Ingegnere del software non è riconosciuta legalmente e non esiste un albo professionale.

E' in corso, a livello mondiale, un'iniziativa per definire la professione e creare le basi per un albo degli Ingegneri del software. Alla professione sono richieste competenze specifiche, regole comportamentali ed etiche e riconoscimenti legali.

Alla nuova professione è richiesta una competenza di base identificata come "Software Engineering Body of Knowledge - SWEBOK", da acquisire presso università o scuole di specializzazione.

La metodologia proposta riprende le 10 competenze identificate:

1. Software Requirements;
2. Software Design;
3. Software Construction;
4. Software Testing;
5. Software Maintenance;
6. Software Configuration Management;
7. Software Engineering Management;
8. Software Engineering Process;
9. Software Engineering Tools and Methods;
10. Software Quality.

Ciascuna area di competenza indirizza il relativo processo, i metodi, le tecniche e gli strumenti a supporto.

5.2 Processi maturi ...

Il processo (maturo) è uno dei tre pilastri su cui si basa una metodologia anch'essa matura.

Gli altri due pilastri, come già detto, sono "la competenza delle persone" e "i metodi, le tecniche e gli strumenti a supporto".

5.2.1 Che cos'è un processo maturo?

Un processo è considerato maturo quando permette di raggiungere gli obiettivi.

In un progetto software il processo adoperato deve permettere di:

- sviluppare software di qualità;
- controllare i tempi di realizzazione;
- controllare i costi del progetto.

Una metodologia all'altezza delle attuali esigenze del mercato utilizza un processo di sviluppo software aderente allo standard ISO 9001:2000 ed adotta il modello CMMI come modello di maturità dei processi.

Secondo la letteratura un processo diviene maturo quando esso è definito, documentato, conosciuto, utilizzato da tutti, valutato nei suoi risultati e modificato per rispondere sempre meglio alle esigenze dell'organizzazione.

Utilizza inoltre le migliori pratiche (best practice) disponibili: Sviluppo iterativo del software; Gestione dei requisiti; Progettazione centrata sull'utente; Revisione tecnica; Verifica della qualità del software; Controllo delle modifiche e della configurazione; Utilizzo di metriche del software e del processo.

Il processo deve potersi utilizzare sia per la manutenzione di applicazioni tradizionali (legacy) sia per lo sviluppo di applicazioni tecnologicamente attuali (object-oriented, tecnologie Web, ecc.).

5.2.2 Le fasi del ciclo di vita ...

Le fasi coprono l'intero ciclo di vita del software:

- Studio di fattibilità;
- Analisi e progettazione;
- Codifica e test unitario;
- Testing e collaudo;
- Rilascio in esercizio;
- Manutenzione correttiva ed evolutiva.

5.2.3 La gestione del progetto ...

Il processo di sviluppo include quello per la gestione del progetto. Quello suggerito è basato sulla metodologia apposita Project Management Methodology, conosciuta come "Project Management Body Of Knowledge" (PMBOK) definita dal Project Management Institute (PMI) che racchiude la migliore esperienza maturata sul campo a livello internazionale.

Il processo per la gestione del progetto di sviluppo include le seguenti aree gestionali:

- Pianificazione e controllo del progetto;

- Gestione del rischio;
- Gestione delle modifiche e della configurazione;
- Gestione della qualità;
- Gestione della documentazione;
- Gestione delle forniture.

5.2.4 Livello di dettaglio ...

Il livello di dettaglio cui giungere nella descrizione di un processo dipende dal livello culturale dell'ambiente cui è dedicato. In un ambiente culturalmente elevato quale quale che si presuppone adottati la metodologia proposta, i dettagli operativi non sono necessari.

Senza giungere a dettagli minuziosi, ciascuna fase è descritta almeno in termini di:

- Criteri di attivazione e di uscita della fase (Entry and Exit Criteria);
- Prodotti di fase realizzati (deliverable);
- Attività svolte;
- Controlli e verifiche eseguiti;
- Ruoli coinvolti;
- Metodi, tecniche e strumenti adoperati;
- Metriche di prodotto e di processo adoperate;
- Evidenze prodotte.

5.3 Utilizzo di metodi, tecniche, strumenti e metriche efficaci ...

5.3.1 Di cosa si tratta?

Lo sviluppo del software si è molto evoluto negli ultimi anni mettendo a disposizione dei progettisti e degli sviluppatori un grande numero di metodi, tecniche e strumenti.

Ovviamente, non tutto si adatta alle esigenze specifiche dell'organizzazione, del progetto o della tecnologia adoperata. La

maturità di un'organizzazione software sta nello scegliere ed adattare alle proprie esigenze quanto di meglio sia disponibile alla data.

Ma torniamo ai metodi, alle tecniche e agli strumenti a supporto dello sviluppo software.

Un'organizzazione che produce software non può essere efficiente ed efficace senza l'utilizzo di metodi, tecniche e strumenti adeguati (come ogni altra attività specialistica, d'altronde!).

Quali potrebbero essere quelli di maggiore utilità, giusto per iniziare il discorso?

Un modo di classificare i metodi, le tecniche, gli strumenti e le metriche del software è proposto nelle sezioni elencate qui di seguito.

- Metodi e tecniche ...
- Strumenti a supporto dello sviluppo ...
- Metriche del software ...

5.3.2 Metodi e tecniche a supporto dello sviluppo software ...

I "metodi" sono i concetti di base, le linee guida e le "tecniche" che l'ingegneria del software definisce dal punto di vista metodologico per la realizzazione del software.

Senza conoscere i principi di base è difficile scrivere del buon codice!

Vediamo le categorie principali.

Metodi e tecniche:

1. Principi di base
2. Tecniche di sviluppo
3. Tecniche di verifica e validazione
4. Tecniche di controllo della qualità
5. Meta-metodi
6. Stili e pattern

Tecnologie di sviluppo:

1. Linguaggi

Si seguito si fornisce una breve descrizione di ciascun elemento menzionato.

5.3.3 Metodi e tecniche

1. Principi di base

Si tratta dei principi di base da seguire nello sviluppo del software. Possono (e quindi devono) essere utilizzati per progettare, sviluppare e verificare codice con il livello di qualità atteso.

- Modularità del software (information hiding, coesione, disaccoppiamento);
- Robustezza del software (gestione delle condizioni di errore e capacità di resistere al sovraccarico);
- Interoperabilità del software (capacità di operare in connessione con altro software);
- Facilità di evoluzione del software (flessibilità, scalabilità, estendibilità);
- Qualità del software (usabilità, affidabilità, efficienza, manutenibilità, portabilità, soddisfazione del cliente).

Si tratta in parte di una diversa rappresentazione delle caratteristiche del software descritte dalle norme ISO/IEC 9126 e di altri principi basilari tratti dall'ingegneria del software.

2. Tecniche di sviluppo

Si tratta delle diverse metodiche per guidare il processo di sviluppo software.

Dipendono dal diverso livello di formalismo seguito.

- Tecniche di sviluppo "informali" (Metodi agili, eXtreme Programming, Unified Process);
- Tecniche di sviluppo "semi-formali" (Object-Oriented, Data Flow, Metodi strutturati, Domain specific);
- Tecniche di sviluppo "formali" (Model Checking, Logiche temporali).

Le tecniche di sviluppo sono a loro volta supportate da modelli (o meta-modelli) e altri strumenti (Stili e Pattern).

3. Tecniche di verifica e validazione

Si tratta di tecniche per la verifica e la validazione costante della qualità dei prodotti di fase realizzati durante l'intero ciclo di sviluppo.

Revisioni tecniche

La tecnica consiste nell'ispezioni dei documenti prodotti nelle prime fasi del ciclo di sviluppo. Ha lo scopo di ricercare e rimuovere gli errori commessi ed evitare che questi si propaghino (vedi "Teoria della propagazione degli errori nel software") nelle fasi successive e quindi al codice finale.

La revisione dei documenti verifica che essi siano completi, aderenti agli standard, coerenti, corretti e indirizzino tutti i requisiti, espliciti ed impliciti. I documenti da sottoporre a revisione sono quelli tecnici (documento dei requisiti, specifiche funzionali e tecniche, documenti di disegno), i piani (piano di progetto, di test, di qualità, di gestione della configurazione, di rilascio), i documenti di testing (casi di test, matrici di test, scenari di test), gli ambienti di test.

E' bene sottoporre a revisione tecnica anche il codice, in modo da verificare (magari a campione su tutti i programmatori) che esso sia scritto secondo gli standard di programmazione (ovviamente devono essere definiti tali standard!);

Testing

E' l'esecuzione del codice secondo diversi livelli di aggregazione (test unitario, test d'integrazione, test di sistema, test d'accettazione).

Ciascun tipo di test fa uso di particolari tecniche (white-box, black-box, error-guessing).

A livello di integrazione dei singoli moduli in componenti le tecniche adoperate sono due (top-down e bottom-up).

L'integrazione del codice utilizza la tecnica dei componenti "fantasma" (scaffolding) per sostituire quelli ancora no disponibili; questi possono essere di due tipi (driver e stub);

4. Tecniche di controllo della qualità

Si tratta di tecniche per la pianificare e controllare la qualità del prodotto software.

In estrema sintesi, il software è di qualità quando "fa quello che deve fare" e "lo fa bene".

In altre parole, il software è di qualità quando "sviluppa tutti i requisiti" ed è "privo di difetti".

La prima caratteristica (il software sviluppa tutti i requisiti) è verificata tramite le revisioni tecniche interne ed i test eseguiti. Le due tecniche

sono state descritte nel punto precedente (3. Tecniche di verifica e validazione).

La seconda caratteristica (il software è privo di difetti) è anch'essa verificata durante l'esecuzione dei test rilevando i difetti e correggendo gli errori.

Il problema è che la caratteristica si riferisce ai difetti che gli utenti rileveranno durante l'utilizzo del prodotto dopo il suo rilascio in esercizio.

La domanda corretta è dunque: "Quanti difetti sono presenti nel prodotto rilasciato in esercizio?".

Il calcolo è fatto utilizzando le tecniche descritte qui di seguito.

Defect removal profile

Si tratta di una tecnica che permette di definire il profilo di difettosità del prodotto software.

La tecnica suggerisce di registrare il numero di difetti rilevati in ciascuna fase del ciclo di sviluppo: Analisi, Disegno, Codifica e UT, Test d'integrazione, Test di sistema, Test d'accettazione.

Nelle fasi alte del ciclo di vita i difetti sono quelli rilevati e corretti con le revisioni tecniche interne (ispezioni). Nelle fasi successive sono le attività di test a rilevare i difetti.

La tecnica si basa sull'osservazione che i difetti rilevati si distribuiscono secondo una curva a campana. La forma e l'altezza della curva dipende da vari fattori tra cui i più significativi sono la dimensione del prodotto, la tecnologia utilizzata, la complessità del software, il processo di sviluppo e la competenza delle persone.

Solo la registrazione dei dati in un archivio dei progetti permette di avere a disposizione dati sufficienti per pianificare la curva di rimozione dei difetti di un prodotto da sviluppare.

La forma e l'altezza della curva a consuntivo permette di estrapolare la difettosità residua del software rilasciato in esercizio.

Testing defect removal

Tecnica di controllo del livello di saturazione dei test ... (IN FASE DI COMPLETAMENTO)

Defect propagation

"Teoria della propagazione degli errori nel software" (IN FASE DI COMPLETAMENTO)

5. Meta-metodi

Si tratta di modelli non realizzati per guidare lo sviluppatore nella realizzazione del software ma per suggerire all'organizzazione come modellare i propri processi produttivi e gestionali in ottica di miglioramento continuo. Alcuni modelli di rilievo sono:

- Il modello GQM (Goal Question Metric) rappresenta un modello strutturato per la creazione di un sistema di metriche;
- Il modello CMMI (Capability Maturity Model Integration) costituisce un modello per il miglioramento dei processi produttivi e di quelli gestionali.

6. Stili e Pattern

Rappresentano i diversi stili con cui lo sviluppatore si abitua a risolvere i problemi e costruire la propria esperienza. Alcuni di questi modelli risultano particolarmente efficaci e diventano materia da condividere e copiare (o almeno seguire nelle loro linee principali).

5.3.4 Tecnologie di sviluppo

Le tecnologie di sviluppo sono quelle più conosciute e seguite. Spesso (purtroppo) si tende a compendiare in esse la totalità dei metodi e tecniche. Principalmente sono costituite dai linguaggi di programmazione, dagli strumenti di sviluppo e dagli ambienti di sviluppo.

1. Linguaggi

Si tratta dei comuni (ma per questo non banali) linguaggi utilizzati per lo sviluppo del software.

Sono in grande numero e coprono le diverse esigenze applicative e tecnologiche.

Si sono profondamente evoluti nel tempo - prima, seconda, terza, quarta, quinta e sesta (?) generazione - per meglio sfruttare le potenzialità delle nuove tecnologie.

I **linguaggi di programmazione** sono quelli utilizzati per la produzione del codice; tra quelli più attuali ricordiamo: Java, C e C++, C#, ecc.

I **linguaggi di mark-up** sono utilizzati per descrivere informazioni e concetti in ambiente Internet e tecnologia Web. Tra quelli più conosciuti ricordiamo: HTML, XML, ecc.

I **linguaggi di modellazione** sono utilizzati per tradurre in termini più formali quanto concordato con il cliente (requisiti e scenari di utilizzo) e l'architettura del sistema. Indirizza quindi i due domini: quello del problema e quello della soluzione. Il linguaggio di modellazione più conosciuto (ed utilizzato) è l'UML (Unified Modelling Language).

I **linguaggi domain-specific** sono invece utilizzati per specifici ambienti o domini. Un esempio tipico è il linguaggio SQL per l'interrogazione delle base dati.

5.4 Strumenti a supporto dello sviluppo

5.4.1 Strumenti di sviluppo

Si tratta degli strumenti (tool) che supportano lo sviluppatore nelle diverse fasi del ciclo di produzione del software. Si basano su uno o più dei linguaggi menzionati sopra. Si dividono a seconda della fase del ciclo di sviluppo in cui si applicano:

- **Strumenti per l'analisi:** sono utilizzati per descrivere in linguaggio formale (es. UML) il dominio del problema e quello della soluzione (es. Rational Rose, MS Visio, ecc.);
- **Strumenti per lo sviluppo (programmazione):** sono utilizzati per scrivere il codice, eseguire il test unitario (debugging) e correggerlo;
- **Strumenti per il testing:** sono utilizzati per la progettazione, l'esecuzione ed il controllo dei test (es. Rational TestManager, Test Director di Mercury, ecc.). Alcuni strumenti permettono la registrazione dei test eseguiti e la successiva esecuzione automatica; altri permettono di simulare particolari condizioni di carico e stress del sistema;
- **Strumenti per il reverse engineering:** sono utilizzati per costruire informazioni mancanti partendo da codice esistente necessarie in fase di manutenzione di applicativi esistenti;
- **Strumenti per la gestione della configurazione:** sono utilizzati come supporto alla gestione del prodotto (componenti, sottocomponenti, elementi unitari) e loro versioni, alla gestione delle modifiche ed alla costruzione dei rilasci.

5.4.2 Ambienti di sviluppo

Sempre più spesso gli strumenti di sviluppo sono integrati in un unico prodotto che copre più fasi del ciclo di sviluppo. Le funzionali sono così integrate, le informazioni condivise in un unico DB e le interfacce risultano omogenee e di più facile utilizzo (es. JBuilder, Suite Rational, Suite Borland, ecc.).

Nota: gli ambienti open source forniscono molti ed ottimi strumenti, anche integrati tra di loro, che coprono l'intero ciclo di vita del software.

5.4.3 Supporto ai processi di sviluppo

Si tratta di strumenti ed ambienti particolarmente sofisticati e complessi che forniscono un valido supporto ai processi di gestione e di sviluppo del software. Il processo di sviluppo è a sua volta un insieme a volte complesso di attività condivise tra più progettisti e sviluppatori. Tra questi strumenti e ambienti ricordiamo:

- **Strumenti di Project Management:** sono utilizzati dal capo progetto per la pianificazione ed il controllo delle attività e dei tempi di realizzazione, per la stima ed il controllo dell'effort, per la produzione dei resoconti (es. MS Project, ecc.);
- **Strumenti di Work Flow:** sono utilizzati realizzare flussi di lavoro semplici o molto complessi orchestrando le diverse attività previste dal processo automatizzato;
- **Strumenti per la gestione del sistema (System Managment):** sono utilizzati per la gestione delle applicazioni, per monitorare le prestazioni, intercettare le condizioni anomale, generare segnali di "alert" e permettendo di intervenire per recuperare le situazioni critiche;
- **Altri tipi di strumenti:** esistono altri strumenti difficile da elencare senza diventare noiosi.

5.5 Metriche del software ...

5.5.1 Perché misurare?

Le metriche del software meritano un discorso particolare.

"Senza misure non è possibile alcun miglioramento!" ha detto perentoriamente qualcuno.

E' un'asserzione importante quanto assoluta.

Molti dei problemi riscontrati nei progetti software in crisi sono ascrivibili alla mancanza di una cultura della misurazione. Di quali misurazioni stima parlando? Di quelle fondamentali per un progetto software: il dimensionamento del software da sviluppare, la produttività dell'organizzazione, la stima dei tempi e dei costi di realizzazione, la difettosità del software.

La carenza di misurazioni è direttamente associata alla "scarsa accuratezza delle stime", un altro bel problema!

"In questa azienda i consuntivi non rispettano mai i preventivi!" tuona la direzione.

Ma erano le stime iniziali ad essere sbagliate o è la gestione non corretta del progetto a generare ritardi ed extra costi? Bella domanda. Forse sono strettamente legate tra di loro le due cose.

Stime corrette ed affidabili richiedono la presenza in azienda di una "cultura delle misure" consolidata. E questo genera un effetto positivo in tutta l'organizzazione: anche i capi progetto hanno questa cultura e quindi gestiscono i progetti adottando le metriche necessarie. Ritardi ed extracosti sono subito individuati e corretti.

Ma in una tale organizzazione anche i processi sono misurati per verificare se questi siano efficaci o meno. E se non lo sono (e solo le misure possono dirlo) vengono opportunamente migliorati.

Un'organizzazione con tale cultura riesce anche a misurare se stessa e a capire dove può migliorare.

Per finire, un'organizzazione che adoperi sistematicamente le misure (non troppe, solo quelle che servono!) non può che far bene e migliorare di continuo. Chi invece non le usa ...!

5.5.2 Cosa misurare?

Si tratta di metodi e tecniche per misurare "cosa" si realizza (il prodotto), "come" lo si realizza (il processo, il progetto) e "chi" lo realizza (l'organizzazione).

Le metriche coprono quindi l'intera organizzazione software: l'**organizzazione** stessa coinvolta, il **progetto** software, il **processo** di sviluppo, il **prodotto** software sviluppato.

5.5.3 Come misurare?

Il processo relativo alle misurazioni prevede le seguenti attività:

- **Definire le metriche necessarie:** si tratta della definizione delle metriche necessarie per valutare la qualità dei processi, dei progetti e dei prodotti realizzati partendo dagli obiettivi e selezionando gli indicatori più efficaci;
- **Pianificare il programma di misurazione:** si tratta della pianificazione del programma di misure da realizzare (cosa, come, chi e quando misurare) e le modalità di reporting e di valutazione. Il piano delle misure è documentato nel Piano della qualità;
- **Eseguire il programma di misurazione:** si tratta dell'esecuzione delle misurazioni pianificate durante il ciclo di sviluppo, della produzione della reportistica, della valutazione dei risultati e della predisposizione di eventuali azioni necessarie in caso di scostamento dai valori attesi.

5.5.4 Alcuni esempi di metriche del software

Di seguito sono proposte alcune metriche semplici per indirizzare gli obiettivi dell'azienda. L'esempio riportato è reale ed è stato sperimentato con successo da un gruppo di piccole e medie aziende italiane di software.

Gli obiettivi che si intendono misurare con le metriche adottate sono riportate qui di seguito.

Tabella 3. Obiettivi di un'organizzazione software.

Obiettivo	Descrizione
O1	Capacità dell'organizzazione di far fronte alle esigenze dei progetti garantendo processi, mezzi e risorse adeguati;
O2	Capacità dei progetti di rispettare i tempi, i costi ed il livello di qualità atteso;
O3	Capacità di pianificare i progetti con stime corrette e realistiche;
O4	Capacità di controllare puntualmente lo stato di andamento dei progetti e di reagire prontamente nei casi di deviazione dai piani;
O5	Capacità dell'organizzazione di sviluppare prodotti con la qualità attesa;
O6	Capacità di migliorare i processi per aumentare le capacità descritte sopra.

Un esempio concreto di come tenere sotto controllo gli obiettivi aziendali è riportato qui di seguito.

Un esempio concreto

Supponiamo di aver definito gli obiettivi aziendali riportati in precedenza.

Le metriche adoperate per tenere sotto controllo gli obiettivi aziendali di cui sopra sono quelle riportate qui di seguito. Sperimentate con successo, possono essere adattate alle esigenze particolari di ciascuna azienda che le voglia utilizzare. Per motivi di semplicità l'indicatore (cosa misurare) è espresso in linguaggio naturale. Gli algoritmi di misurazione possono essere ricavati facilmente da ogni esperto di metriche.

Tabella 4. Dettagli sulle metriche relative agli obiettivi di un'organizzazione software.

Categoria: Metriche relative all'organizzazione (O1)

Metrica	Cosa si misura
Adeguatezza della struttura organizzativa alle esigenze dei progetti (disponibilità delle risorse richieste)	Disponibilità delle risorse pianificate (<i>Numero e Permanenza di persone nel progetto</i>).
	Adeguatezza delle risorse disponibili (<i>Competenza, Esperienza, Produttività rispetto a quelle richieste</i>).
	Impatti sul cliente causati dalle persone (<i>Numero di rilievi fatti dal cliente</i>).

Categoria: Metriche di processo (O6)

Metrica	Cosa si misura
Aderenza al processo (da parte dei progetti)	Livello di aderenza del progetto ai processi aziendali (<i>Numero di non conformità rilevate nei progetti</i>).
Efficacia dei processi	Livello di efficacia dei processi (<i>Numero di non conformità relative ad obiettivi dei progetti non raggiunti e da addebitare ai processi stabiliti</i>). Esempio: Metodi di stima non adeguati, Dati statistici incompleti o errati, Checklist incomplete o errate, Template non adeguati, Metodi e tecniche non adeguate, Strumenti non adeguati, ecc.).

Categoria: Metriche di progetto (O2, O4)

Metrica	Cosa si misura
Rispetto dei tempi di consegna (O2, O4)	Ritardo nelle consegne (<i>Numero di giorni di ritardo rispetto ai piani delle consegne</i> ³).
Contenimento dei costi (O3, O4)	Extracosti (<i>Scostamento percentuale dei costi a consuntivo rispetto ai preventivi</i> ⁴).
Raggiungimento della qualità (O4, O5)	Rilascio dei deliverable previsti (<i>Numero di deliverable rilasciati rispetto a quelli previsti</i>). Qualità dei deliverable (<i>Numero di rilievi fatti dal cliente in fase di consegna, approvazione, collaudo</i>)

Categoria: Metriche di prodotto nelle varie fasi del ciclo di vita (O5)

Metrica	Cosa si misura
- Analisi: Completezza e correttezza dell'interpretazione dei requisiti	Qualità dei requisiti (<i>Numero di errori rilevati durante l'intero ciclo di sviluppo ed imputabili ad una incompleta o non corretta interpretazione dei requisiti</i>).
- Progettazione: Qualità della progettazione	<i>Livello di strutturazione dell'architettura applicativa.</i> <i>Livello di coesione dei componenti applicativi.</i> <i>Livello di disaccoppiamento dei componenti applicativi.</i> <i>Livello di strutturazione dei dati (information hiding).</i> <i>Livello di aderenza agli standard tecnologici richiesti (OO, CORBA, SOA, SOAP, ecc.).</i>

³ E' importante analizzare le cause dei ritardi per verificare se il problema sia stato generato da stime iniziali non corrette, da una pianificazione non realistica o da una conduzione del progetto non adeguata.

⁴ Nel caso di scostamento dei costi a consuntivo rispetto alle stime iniziali occorre distinguere se esso (lo scostamento) sia dovuto ad una errata stima iniziale oppure ad uno "sforamento" vero e proprio del budget.

Qualità del software	- Codifica:	<i>Livello di aderenza del codice agli standard di codifica (strutturazione, commenti, dimensioni, ecc.).</i>
Difettosità residua (del software rilasciato in esercizio)	- Rilascio:	<i>Numero di difetti rilasciati in produzione/Dimensioni del software (FP o KLocs)⁵.</i>
Qualità della documentazione prodotta (documenti tecnici e manuali)	- Tutte le fasi:	<i>Completezza della documentazione (documenti prodotti rispetto a quelli previsti).</i>
		<i>Aderenza agli standard documentali applicabili (numero di anomalie rilevate).</i>
		<i>Correttezza della documentazione (numero di anomalie rilevate).</i>
Efficacia dei test	- Collaudo:	<i>Livello di copertura dei test rispetto ai requisiti (Matrice di tracciabilità dei requisiti).</i>
		<i>Numero di errori rilevati rispetto al numero previsto (ricavato dalla curva di rimozione degli errori).</i>
		<i>Numero medio di errori rilevati per caso di test.</i>
Efficienza dei test (produttività)		
		<i>Giorni persona spesi per caso di test eseguito</i>

⁵ Il numero dei difetti (errori) residui nel software rilasciato in esercizio è calcolato estrapolandolo dall'andamento della curva di rimozione dei difetti nelle fasi del ciclo di sviluppo. La rimozione degli errori è effettuata tramite le revisioni tecniche interne prima (fasi alte del ciclo) e con le attività di test dopo. L'andamento della curva mostra infatti quale sarebbe il numero di errori rimossi nella fase successiva a quella dei test e del collaudo finale: tale fase coincide proprio con l'utilizzo del software da parte degli utenti. Presupposto per l'utilizzo delle curve è quella della registrazione dei difetti rilevati in tutte le fasi del ciclo di sviluppo ed una corretta progettazione ed esecuzione dei test.

Bibliografia

Di seguito è riportata la bibliografia di base cui l'autore fa riferimento per le sue attività. Essa è distinta per argomento: Qualità, ISO9000, CMMI, Sviluppo professionale, Ingegneria del software ecc. Essa non pretende di essere esaustiva né definitiva; rappresenta invece una buona base di partenza per approfondire i temi trattati nel libro.

Ingegneria del software

- [PLATT07] David S. Platt - 2007
Perchè il software fa schifo... – Mondadori Informatica
- [FUGGETTA06] Annalisa Binato, Alfonso Fuggetta, Laura Sfardini - 2006
Ingegneria del software, Pearson – Addison Wesley
- [JOEL05] Joel Spolsky - 2005
Joel e il Software, Apress
- [SOMMERV05] Ian Sommerville - 2005
Ingegneria del software, PEARSON Addison-Wesley
- [PRESSMAN05] Roger S. Pressman – 2005
Principi di Ingegneria del Software, McGraw-Hill
- [GHEZZI04] Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrili - 2004
Ingegneria del software, PEARSON Prentice Hall
- [KUCHTEN99] Philippe Kruchten - 1999
Unified Process Introduction, Addison-Wesley
- [CICOINI96] Vincenzo Ambriola, Giovanni Cicogni - 1996
Laboratorio di Programmazione, Jackson Libri
- [BROOKS95] Brooks Frederick P. - 1995
The Mythical Man-Month: Essays on Software Engineering – 20th Anniversary Edition, Addison-Wesley, Reading(MA)
- [MARTIN90] James Martin - 1990
Information Engineering Books: I (Introduction), II (Planning and Analysis), III (Design and Construction), Prentice-Hall
- [NATO68] *Atti della conferenza Nato tenuta nel 1968*

Project Management

- [NOKES05] Nokes Sebastian – Greenwood Alan – 2005
Il Project Management – Tecniche e processi, Prentice Hall, Fianciual Times
- [PMBOK04] Project Management Institute (PMI) – 2004
A Guide to the Project Management Body of Knowledge (PMBOK® Guide), Third Edition, Project Management Institute – Pennsylvania USA
- [KLIEM04] Ralph L. Kliem – 2004
Leading High-Performance Projects, J. Ross Publishing
- [ARCHIBALD97] Russel D. Archibald – 1997
Project Management, La gestione di progetti e programmi complessi, Franco Angeli
- [MEREDITH95] Jack R. Meredith, Samuel J. Mantel Jr. – 1995
Project Management. A Managerial Approach, Third Edition, John Wiley & Sons

Testing

- [MOSLEY03] Daniel J. Mosley, Bruce A. Posey - 2003
Collando del Software (titolo originale: Just Enough Software Test Automation), McGram Hill
- [KANER03] Cem Kaner, Lames Bach, Bret Pettichord – 2003
Lessons Learned in Software Testing, A Context-Driven Approach, Wiley
- [CABExx] Mc Cabe T.J., Watson A.R., 19xx
Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric, Dolores R. Wallace, Editor
<http://www.mccabe.com/?file=./prod/nist/Index.html>
- [CICOONI98] Cignoni G. A., De Risi P. - 1998
Il test e la qualità del software, Il Sole 24 Ore
- [GILB93] Gilb Tom and Graham Dorothy - 1993
Software Inspection, Addison Wesley
- [BEIZER90] Beizer B. - 1990
Software System Testing and Quality Assurance, International Thomson Computer Press
- [MYERS79] Myers G. J. - 1979
The art of Software Testing, John Wiley & Sons, New York

Metriche del software

- [GUFPI06] GUFPI – ISMA 2005
Metriche del software – Esperienze e ricerche, FrancoAngeli
- [BUGLIONE03] Luigi Buglione –2003
Misurare il software, FrancoAngeli
- [NATALE96] Domenico Natale - 1996
Qualità e quantità nei sistemi software – Teoria ed esperienze
Franco Angeli/Informatica
- [PULFORD96] Kevin Pulford, Annie Kuntzmann, Stephen Shirlaw - 1996
A quantitative approach to Software Management – The AMI Handbook (AMI ESPRIT), Addison-Wesley

Qualità

- [LEGA06] Riccardo Lega - 2006
La Patente Europea della Qualità (EQDL)
FrancoAngeli
- [LEONARDI00] Erika Leonardi - 2000
Capire la qualità – ISO9000 – Tutto quello che occorre capire per applicare con profitto le nuove norme
Il Sole 24 Ore
- [BARBARINO98] Filippo C. Barbarino, Erika Leonardi - 1998
ISO9000 Sistema Qualità e Certificazione
Il Sole 24 Ore
- [PINDER96] Mark Pinder & Stuart McAdam - 1996
La consulenza interna
Jackson Libri
- [MUNRO94] Lesly Munro-Faure, Malcom Munro-Faure - 1994
Qualità totale: tecniche di attuazione
Jackson Libri
- [PALA94] Giorgio Pala - 1994
La qualità. Perché, per chi e come farla
Franco Angeli
- [BANC193] Alessandro Banci, Giuseppe Iacono - 1993
La qualità nei progetti software
Franco Angeli

- [NOCENTINI93] Stefano Nocentini - 1993
Il sistema di qualità del software
ETASLIBRI
- [WHEELER93] Donald J. Wheeler - 1993
Understanding Variation – The Key To Managing Chaos
SPC Press
- [ZEITHAML91] Valerie A. Zeithaml, A. Parasuraman, Leonard L. Berry - 1991
Servire qualità
McGraw-Hill
- [RUMMLER90] Geary A. Rummler and Alan P. Brache - 1990
Improving performance
Jossey-Bass Publishers
- [CROSBY86] Philip B. Crosby - 1986
La qualità non costa
McGraw-Hill

ISO9000

- [ISO9001:2000] ISO/IEC 9001:2000
Quality Management Systems – Requirements
- [ISO9000:2000] ISO/IEC 9000:2000
Quality Management Systems – Fundamentals and Vocabulary
- [ISO9004:2000] ISO/IEC 9004:2000
Quality Management Systems – Guidelines for performance improvement
- [ISO9003:2004] ISO/IEC 9003:2004
Software and Systems Engineering – Guidelines for the application of ISO 9001:2000 to computer software
- ISO9126-1:2001] ISO/IEC 9126-1:2001
Software engineering – Product Quality Part 1: Quality Model
- [ISO12207:95] ISO/IEC 12207:1995
Information Technology – Software Lifecycle Processes
- [ISO12207:02-1] ISO/IEC 12207:1995/Amd.1:2002
Information Technology – Software Lifecycle Processes-Amendment 1
- [ISO15271:98] ISO/IEC TR 15271:1998
Information Technology – Guide for ISO/IEC 12207 Software Lifecycle Processes)

- [ISO16326:99] ISO/IEC TR 16326:1999
Software Engineering – Guide for the application of ISO/IEC 12207 in Project Management

CNIPA

- [CNIPA26] iQuaderni, N. 26. aprile 2006
Linee guida sulla qualità dei beni e servizi ICT per la definizione e il governo dei contratti della P.A. Modelli per la Qualità delle Forniture ICT. Manuale di riferimento. CNIPA
- [CNIPA27] iQuaderni, N. 27. maggio 2006
Linee guida sulla qualità dei beni e servizi ICT per la definizione e il governo dei contratti della P.A. Governo dei contratti ICT. Manuale Applicativo. CNIPA

AICQ-CI

- [AICQCI11] Quaderno N. 11, Aprile 2001
Linee guida per l'applicazione della Norma ISO 9001:2000 (Vision 2000) per lo sviluppo e la manutenzione del software. Mario Cislighi, AICQ-CI.

Perché molti progetti software falliscono? L'argomento, vecchio quanto l'informatica stessa, purtroppo è sempre d'attualità. Il numero di progetti software che ogni anno falliscono nel mondo, per vari motivi, è sempre molto alto. Analisi sulle cause dei fallimenti dei progetti software sono state effettuate più volte, da professionisti e studiosi, anche autorevoli. Le conclusioni sono, ahinoi!, impietose: nonostante l'ingegneria del software produca metodologie sempre più adatte alle nuove tecnologie ed alla complessità dei progetti, molti di questi non raggiungono gli obiettivi: ritardi nei tempi di consegna, costi superiori ai piani, qualità inferiore alle attese. Il male sembra inguaribile, la causa cronica: "scarsa maturità delle organizzazioni software". Che significa allora maturità di un organizzazione software? Significa che tutta l'organizzazione, dall'alta direzione ai livelli tecnici dei programmatori, non possiedono la maturità professionale, ognuno per il ruolo che ricopre, richiesta per svolgere il proprio compito con la qualità richiesta.

Il documento riassume quanto è stato pubblicato (nei libri e sul Web) sul tema del fallimento dei progetti software e delle relative cause. Fornisce anche una soluzione (meglio un approccio ad affrontare la soluzione) basata sull'esperienza.



Ercole Colonese svolge la sua attività di consulente essenzialmente nell'area dello sviluppo applicativo. La sua esperienza è maturata in moltissimi anni di lavoro presso i laboratori internazionali IBM di sviluppo prodotti software dove ha ricoperto ruoli tecnici e manageriali. Ha realizzato numerosi sistemi qualità aziendali certificati ISO9000 presso piccole, medie e grandi aziende, pubbliche e private. Ha implementato modelli di eccellenza (EFQM, Malcom Bauldrige ecc.). Ha condotto diversi progetti di reingegnerizzazione dei processi di sviluppo software in ottica di miglioramento delle performance e della qualità. Ha applicato con successo i modelli di maturità dei processi come il *Capability Maturity Model* del *Software Engineering Institute* (SEI-CMM e CMMI). Come docente, ha tenuto corsi sulla qualità del software e le metodologie di sviluppo presso il Learning Center IBM. Ha pubblicato diversi articoli sulle tematiche dello sviluppo software ed il ruolo consulenziale. Attualmente è professore aggiunto di Ingegneria del software presso l'Università di Roma Tor Vergata.

e-mail: e.colonese@virgilio.it

www.colonese.it