

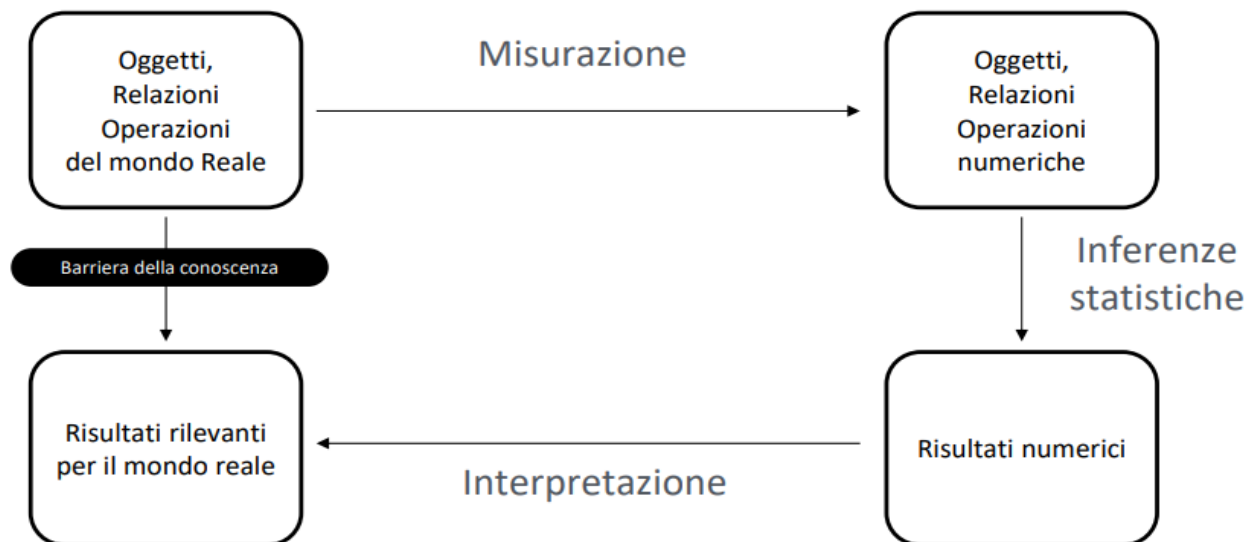
# TEORIA DELLA MISURA

## Perché è importante misurare?

Tra le varie cose, misurare serve a:

- 1) **Caratterizzare**, ovvero estrarre informazioni ben precise su qualche caratteristica di un prodotto.
- 2) **Comparare**, ovvero capire se lo sviluppo di un prodotto A sta andando meglio, peggio o come lo sviluppo di un altro prodotto B con caratteristiche analoghe.
- 3) **Controllare o monitorare** un prodotto.
- 4) **Valutare**, ovvero dare valore a qualche caratteristica di un prodotto, come la sua difettosità sulla base dei dati raccolti durante il suo sviluppo.
- 5) **Predire**, ovvero anticipare il valore che assumerà qualche caratteristica di un prodotto, come ad esempio il suo tempo di consegna.

La misurazione, in breve, ci permette di ottenere dei risultati rilevanti per il mondo reale (come una considerazione, una legge fisica o il risultato di un esperimento) a partire dall'osservazione degli oggetti del mondo reale:



Notiamo che non è possibile avere dei risultati rilevanti per il mondo reale senza ricorrere a un processo di astrazione caratterizzato da operazioni numeriche e risultati numerici e, quindi, senza misurazione. Questo "vincolo" è noto come barriera della conoscenza.

## Definizione

Un **sistema relazionale** consiste in una tupla definita da:

- Un insieme di oggetti.
- Relazioni tra tali oggetti ("è uguale a", "è più di", "è meno di").
- Operazioni binarie tra tali oggetti (somma, sottrazione).

## Definizione

Sia **A** un sistema relazionale di oggetti fisici, sia **B** un sistema relazionale di oggetti formali e sia **m** una misura da A a B (Una mappa 1-1 tra un oggetto fisico ed un oggetto formale, tipo numero).

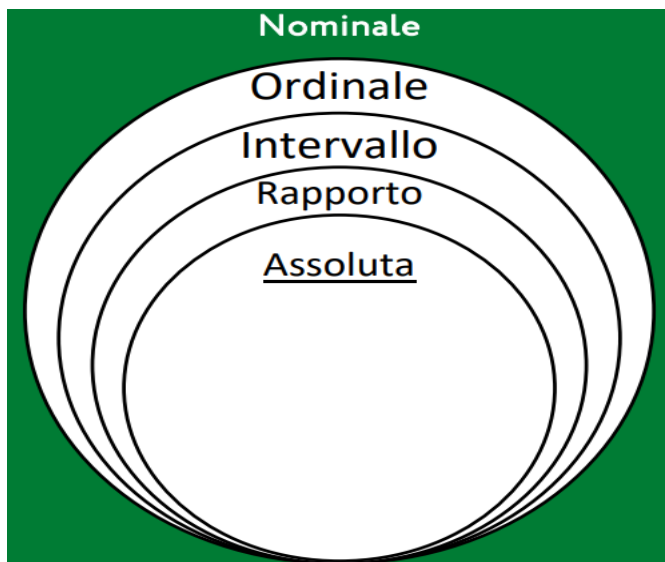
Allora la tupla (A, B, m) è una **scala** se:

- Relazioni in A equivalgono a relazioni in B.
- Ogni operazione in A ha una corrispondente operazione in B.

## Tipi di scale

Tipo	Uso	Relazioni	Operazioni	Esempio
Scala nominale	Dare nomi alle cose.	Uguaglianza	-	Numeri sulle t-shirt dei giocatori di calcio (dove non c'è un numero che ha più valore di un altro).
Scala ordinale	Dare nomi in un ordine specifico.	- Più di - Meno di	-	Podio olimpionico (dove la medaglia d'oro vale di più di quella d'argento ma è impossibile assegnare un valore numerico alle varie medaglie).
Scala intervallo	Assegnare numeri al fine di rendere significativo un intervallo.	Somiglianza, ottenibile tramite relazioni matematiche.	- Mediana - Media	Temperatura su scala Celsius, dove: $\text{Celsius} \times 1,8 + 32 = \text{Fahrenheit}$ Anche voti UNI, che vanno da 0 a 30
Scala rapporto	Effettuare un rapporto tra due misure.	Somiglianza, ottenibile tramite relazioni matematiche.	- Mediana - Media	Lunghezza in metri di un oggetto, che è ottenuta dal rapporto della dimensione dell'oggetto stesso e la dimensione (di 1 metro) della barra di platino-iridio conservata a Sèvres.
Scala assoluta	Trattasi di una scala utilizzata quando si ha un unico modo per misurare oggetti.	Somiglianza, ottenibile dalla funzione identità.	- Mediana - Media	Conteggio (non esistono più modi per contare il numero di oggetti).

Tali scale sono caratterizzate da una relazione di inclusione tra loro:



## Glossario

Termine	Significato
Entità	È un elemento dotato di realtà oggettiva. È indipendente e auto contenuta e ha dei caratteri propri. Inoltre, è di un certo tipo, che può essere ad esempio un prodotto, un processo o una risorsa.
Entità misurabile	È una qualunque entità che può essere misurata. È caratterizzata da attributi misurabili.
Tipo di entità misurabile	Le entità misurabili, a loro volta, possono essere classificate a seconda del tipo. Per esempio, il processo è un tipo di entità misurabile.
Modello di entità	È un'astrazione di un'entità che viene modellata in relazione agli scopi che si hanno, al grado di conoscenza che si ha dei suoi caratteri e agli strumenti di modellazione.
Attributo	È una proprietà di un'entità che: - Concorre a determinare l'eventuale stato e gli eventuali comportamenti dell'entità. - Definisce una qualità di interesse dell'entità.
Attributo misurabile	È una proprietà di un'entità misurabile che, appunto, può essere misurata. Dimensione e produttività sono esempi di attributi misurabili.
Modello di attributo	È un'astrazione di un attributo che, come i modelli di entità, dipende dagli scopi che si hanno, dal grado di conoscenza che ne si ha e dagli strumenti di rappresentazione.
Misurazione	È il processo che si esercita su un'entità (in particolar modo su un suo attributo) con lo scopo di produrre un risultato di misurazione.
Risultato di una misurazione	È un valore associato a un attributo di un'entità a seguito di una misurazione. È tipicamente ma non necessariamente numerico: in particolare, appartiene ai valori ammessi della scala utilizzata.
Misura	È uno strumento (una funzione) che permette di associare attributi misurabili con i valori di una scala. Ad esempio, la misura "numero di requisiti" può essere utilizzata per associare un valore all'attributo misurabile "dimensione" che caratterizza il tipo di entità misurabile "progetto".
Metrica	È un termine utilizzato con diverse accezioni nell'ambito dell'ingegneria del software: - È una particolare grandezza da misurare (come una distanza o una superficie). - È l'insieme delle norme che regolano composizione e strutturazione delle misure. - È l'insieme delle misure di un settore o di un software nel suo complesso.

**NB:** Un attributo può essere:

- **Esterno** se è osservabile direttamente dall'utilizzatore dell'entità (come gli attributi public in Java).
- **Interno** se è osservabile esclusivamente nell'ambito dell'entità presa isolatamente, ma non esternamente (come gli attributi private in Java che, per essere acceduti, richiedono l'invocazione di un apposito metodo getter o setter).

D'altra parte, un attributo può anche essere:

- **Direttamente misurabile** (come la dimensione).
- **Indirettamente misurabile** (come la produttività) se può essere misurato solo mediante il coinvolgimento di altri attributi di base.

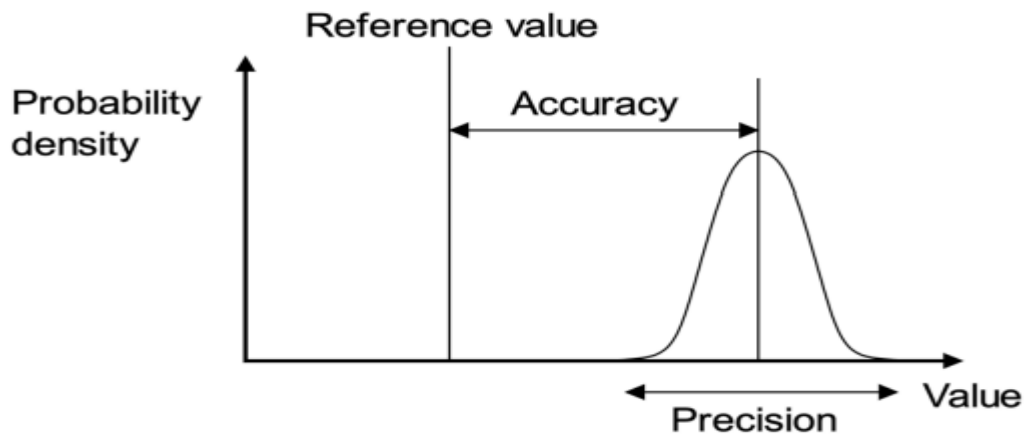
### Caratteristiche della misurazione

- **Influenza:** i sistemi, fra cui le entità che si intendono misurare, sono generalmente modificati e influenzati dalle misurazioni.
- **Ripetibilità:** in analoghe condizioni, le misurazioni dovrebbero dare luogo al medesimo risultato. In altre parole, salvo errori sistematici e/o casuali, una determinata entità, a parità di contesto, stato e attributo misurato, deve essere caratterizzata sempre dalla medesima misura.

- **Errori:** gli errori di osservazione possono essere sistematici (che sono relativi alla differenza tra il valore reale della grandezza in esame e il valore assunto dal risultato della misurazione effettuata su di essa) oppure casuali (che possono essere dovuti a errori di lettura dello strumento di misurazione o di trascrizione del risultato della misurazione).

Un insieme di data point (= risultati di misurazione) derivanti da misurazioni ripetute della stessa quantità si definisce:

- **Accurato (ISO Trueness)** se la media del valore dei data point è vicina al vero valore della quantità misurata.
- **Preciso** se i data point sono “vicini” tra loro.



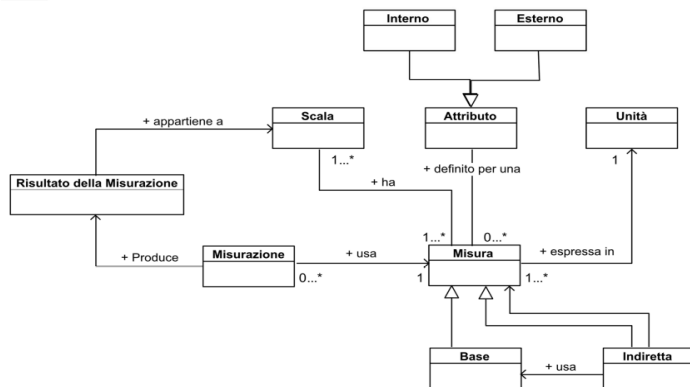
In realtà, l'accuratezza ha un duplice significato poiché ISO la definisce come l'unione tra le due definizioni appena elencate, ovvero come la prossimità dei risultati di misurazione al vero valore della quantità misurata.

### Classic Measurement Ontology (CMO)

La CMO (Classic Measurement Ontology) è un modello di rappresentazione formale secondo cui:

- Un attributo può essere definito per una misura, ovvero può essere associato tramite quest'ultima a un valore appartenente a una determinata scala.
- Un attributo può essere esterno (se è osservabile direttamente a partire da un'entità differente) oppure interno (se è osservabile solo prendendo l'entità isolatamente).
- Ciascuna misura è espressa in un'unità di misura.
- Una misura può essere di base oppure indiretta; quest'ultima usa una o più misure di base.
- Una misurazione fa uso di una misura per produrre il risultato della misurazione.
- Il risultato della misurazione appartiene a una determinata scala, e non è altro che il valore che viene associato a un attributo di un'entità a seguito del processo di misurazione.
- Ciascuna misura ha una o più scale; è possibile effettuare delle trasformazioni da una scala all'altra (dette somiglianze) mediante delle operazioni matematiche.

In conclusione, la CMO può essere rappresentata mediante il seguente diagramma UML:



## FUNCTIONAL & NON FUNCTIONAL SOFTWARE METRICS

### Requisiti non funzionali (NFR)

Sono requisiti che definiscono come dovrebbe essere il sistema. Si suddividono in:

- Operativi
- Revisionali
- Di transizione

Noi ci concentreremo su quelli operativi, tra cui rientrano la disponibilità, la sicurezza e l'usabilità.

### Disponibilità (availability)

È la capacità di un servizio di svolgere la propria funzionalità ogni volta che viene richiesto dall'utente. Si calcola con la seguente formula:

$$Availability\% = \frac{(Tempo\ di\ servizio\ concordato - Tempo\ di\ inattività)}{Tempo\ di\ servizio\ concordato}$$

Dove:

- **Tempo di servizio concordato** = tempo previsto in cui il servizio dovrebbe essere operativo.
- **Tempo di inattività** = quantità di tempo durante il tempo di servizio concordato in cui il servizio non è disponibile.

La disponibilità può anche essere espressa tramite la cosiddetta regola dei nove:

Availability %	Downtime per year	Downtime per month	Downtime per week
90% (one nine)	36.5 days	72 hours	16.8 hours
99% (two nines)	3.65 days	7.20 hours	1.68 hours
99.5%	1.83 days	3.60 hours	50.4 minutes
99.9% (three nines)	8.76 hours	43.8 minutes	10.1 minutes
99.95%	4.38 hours	21.56 minutes	5.04 minutes
99.99% (four nines)	52.56 minutes	4.32 minutes	1.01 minutes
99.999% (five nines)	5.26 minutes	25.9 seconds	6.05 seconds
99.9999% (six nines)	31.5 seconds	2.59 seconds	0.605 seconds
99.99999% (seven nines)	3.15 seconds	0.259 seconds	0.0605 seconds

### Sicurezza (security)

È buona norma utilizzare i **KPI** come tecnica per tenere traccia delle metriche di sicurezza informatica, nonostante si tratti ancora di una pratica non comune e in via di sviluppo.

I KPI (Key Performance Indicator) sono degli indicatori utilizzati per misurare i risultati conseguiti da un'organizzazione. Nell'ambito della sicurezza, i due KPI più importanti sono:

- **MTTD** (tempo medio di rilevamento): è la quantità di tempo mediamente necessaria per identificare un attacco ricevuto.

- **MTTR** (tempo medio di risposta): è la quantità di tempo mediamente necessaria per contenere o risolvere il danno causato dall'attacco ricevuto.

Nonostante la loro importanza, questi due fattori non sempre vengono considerati quanto necessario e, spesso, risultano essere un fattore critico: ad esempio, nel 2017, per le società statunitensi l'MTTD medio è stato di 52 giorni, mentre l'MTTR medio è stato di 208 giorni.

Consideriamo alcune altre metriche di sicurezza rilevanti:

Metrica	Descrizione
Numero di risorse vulnerabili nell'ambiente	È una metrica di sicurezza chiave per determinare i rischi che l'azienda corre e le patch che devono essere effettuate per ridurre tali rischi.
Numero di certificati TLS configurati in maniera errata	Il monitoraggio dei requisiti di sicurezza dei certificati impedisce che essi cadano nelle mani di malintenzionati che possono sfruttare l'identità digitale dell'azienda per rubare le informazioni dei dipendenti.
Volume di dati trasferiti usando la rete aziendale	Se i dipendenti effettuano tanti download (ad esempio di software, video, film e applicazioni) usando la rete aziendale, possono lasciare la porta aperta a malware.
Numero di dipendenti con livello di accesso "super utente"	Per ragioni di sicurezza, è opportuno che i dipendenti accedano solo a dati, risorse e sistemi necessari al proprio lavoro.
Numero di giorni per disattivare le credenziali di un ex dipendente	Idealmente, l'accesso al sistema degli ex dipendenti dell'azienda dovrebbe essere revocato immediatamente.
Numero di porte di comunicazione aperte durante un periodo di tempo	Tutte le porte comuni per i protocolli che consentono sessioni remote (come SSH, Telnet e FTP) devono essere monitorate per un certo periodo di tempo; inoltre, è inopportuno tenere aperte le porte non necessarie, poiché rappresentano solo un punto di ingresso in più per eventuali attacchi o malware.
Frequenza di revisione degli accessi di terze parti alla rete aziendale	Spesso viene concesso l'accesso a terze parti nella rete aziendale per completare un progetto. È importante assicurarsi che l'accesso venga annullato al termine del progetto stesso: in caso contrario, è possibile che la terza parte decida di tornare per estrarre dati o svolgere altre attività dannose; oppure, peggio ancora, se la terza parte viene violata, anche la nostra rete aziendale può essere esposta alla stessa minaccia.
Frequenza di accesso a sistemi aziendali critici da parte di terzi	Per ragioni di sicurezza, è opportuno monitorare i tentativi di accesso a server o applicazioni che non dovrebbero essere presi di mira da utenti non autorizzati.
Percentuale di partner commerciali con politiche di sicurezza informatica efficaci	Se l'azienda investe in sicurezza ma ha partner che non lo fanno, non gode in realtà di alcun servizio di sicurezza.

Una tipologia particolare di metriche di sicurezza prevede solo due valori possibili: **vero** e **falso**.

Tali metriche tipicamente appartengono a un vasto insieme di regole (una check-list) detto **benchmark**. Il risultato del benchmark è normalmente espresso con una **scala intervallo** e rappresenta il valore della "vera metrica" con cui è possibile esprimere il giudizio complessivo sull'entità oggetto di misurazione.

Due esempi di enti che offrono metriche di benchmarking sono:

- **OWASP** (Open Web Application Security Project), che è una fondazione senza scopo di lucro che lavora per migliorare la sicurezza del software.
- **Dipartimento della Difesa degli Stati Uniti** (supportato dal DISA – Defense Information Systems Agency), che prevede uno standard di sicurezza rappresentato da delle check-list che prendono il nome di STIG

(Security Technical Implementation Guide). Gli STIG contengono una guida tecnica per “bloccare” i sistemi informatici e i software che potrebbero altrimenti essere vulnerabili a un attacco.

### Requisiti funzionali

Sono requisiti che definiscono che cosa dovrebbe fare il sistema. Sono più semplici rispetto a quelli non funzionali poiché:

- Sono meno astratti e meno ambigui, per cui normalmente non generano confusione.
- Sono più facili da misurare.
- Sono più facili da testare.
- A differenza dei requisiti non funzionali, possono essere implementati nel sistema in maniera incrementale (ovvero è possibile aggiungere sempre più funzionalità nel tempo, mentre non è possibile aggiungere servizi di sicurezza senza dover riprogettare e re implementare tutto il sistema).

Le metriche orientate alle funzionalità (**functional software metrics**) si concentrano sulla quantità di funzionalità offerte dal software, basandosi sul calcolo del cosiddetto **punto funzione** (FP), che è appunto l'unità di misura che quantifica la funzionalità aziendale fornita dal prodotto (ma non include gli sforzi relativi ai requisiti non funzionali).

### Glossario

Termine	Significato
Processo Elementare (EP)	È la più piccola unità di requisito funzionale che: <ul style="list-style-type: none"><li>- È significativo per l'utente.</li><li>- Costituisce una transazione completa.</li><li>- È autonomo e lascia l'attività dell'applicazione conteggiata in uno stato coerente.</li></ul>
Funzioni dati (data functions)	Sono costituite da risorse interne ed esterne che influiscono sul sistema, come i file logici interni (ILF) e i file di interfaccia esterna (EIF).
Funzioni di transazione (transactional functions)	Sono costituite dai processi di interscambio tra l'utente, le applicazioni esterne e l'applicazione da misurare. Esistono tre tipologie di funzioni di transazione: <ul style="list-style-type: none"><li>- Input esterni (EI).</li><li>- Output esterni (EO).</li><li>- Richieste esterne (EQ).</li></ul>
File logico interno (ILF)	È un gruppo identificabile dall'utente di dati logicamente correlati che risiedono interamente all'interno del boundary dell'applicazione. Il suo intento principale è conservare i dati attraverso uno o più processi elementari dell'applicazione stessa.
File di interfaccia esterna (EIF)	È un gruppo identificabile dall'utente di dati logicamente correlati che risiedono interamente al di fuori del boundary dell'applicazione (poiché mantenuti in un ILF da un'altra applicazione). Per ottenere i dati in un EIF, è necessario sviluppare un'interfaccia apposita.
Input esterno (EI)	È una funzione transazionale in cui i dati <b>entrano</b> nell'applicazione a partire dall'esterno (ad esempio, a partire da una schermata di immissione dati o da un'altra applicazione).
Output esterno (EO)	È una funzione transazionale in cui i dati <b>escono</b> dal sistema per essere inviati all'utente oppure ad altre applicazioni.
Richiesta esterna (EQ)	È una funzione transazionale con componenti di input e output che determinano il recupero dei dati. Di fatto, una richiesta può essere effettuata: <ul style="list-style-type: none"><li>- Da parte dell'utente verso il sistema.</li><li>- Da parte di applicazioni esterne verso il sistema.</li><li>- Da parte del sistema verso l'utente.</li><li>- Da parte del sistema verso le applicazioni esterne.</li></ul>

Record Element Type (RET)	È il più grande sottogruppo di elementi identificabile dall'utente all'interno di un ILF o un EIF.
Data Element Type (DET)	È il sottogruppo di dati all'interno di un File Type Referenced. Sono univoci e identificabili dall'utente.
File Type Referenced (FTR)	È il più grande sottogruppo identificabile dall'utente all'interno dell'input esterno, output esterno o richiesta esterna a cui si fa riferimento.

#### **Processo di conteggio dei FP**

- 1) Determinare il tipo di conteggio.
- 2) Determinare il limite del conteggio.
- 3) Identificare ogni Processo Elementare richiesto dall'utente.
- 4) Determinare i Processi Elementari unici.
- 5) Misurare le data function.
- 6) Misurare le transactional function.
- 7) Calcolare la dimensione funzionale (conteggio dei punti funzione non aggiustato).
- 8) Determinare il fattore di aggiustamento del valore (VAF).
- 9) Calcolare il conteggio dei punti funzione aggiustato.

NB: I passaggi 8 e 9 di questo algoritmo sono facoltativi.



## APPLICARE FP AL SOFTWARE DOCUMENTATO CON UML

### Regole principali

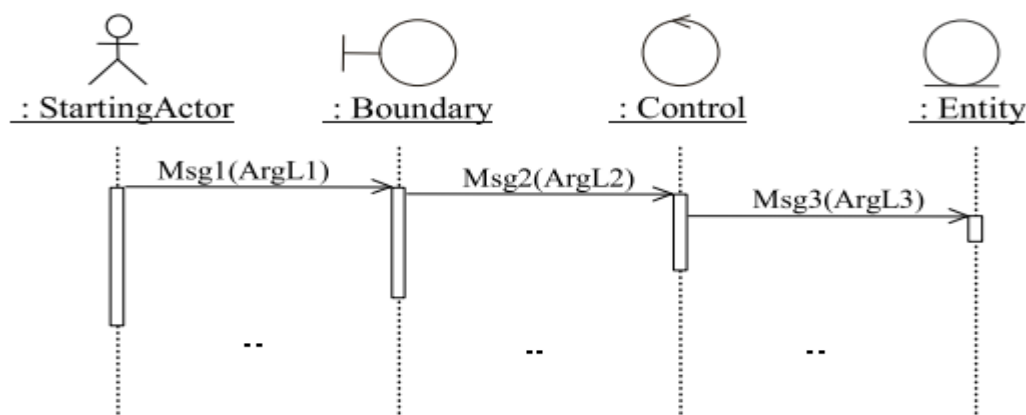
Elemento	Calcolo da effettuare
Attore umano	È sorgente / destinazione di funzioni transazionali.
Attore non umano	È sorgente / destinazione di funzioni transazionali.
Altro tipo di attore (e.g. stampante, timer)	Deve essere scartato dall'analisi dei function point.
Associazione tra caso d'uso e attore	È candidato per alcune funzioni transazionali (e.g. EI, EO, EQ).
Caso d'uso di estensione o di inclusione	È candidato per alcune funzioni transazionali (e.g. EI, EO, EQ).
Classe Entity	È candidata per ILF o EIF.
Classe non Entity	Viene sempre scartata (per cui non rientra nei casi esposti dopo).
Classe astratta	Ciascuna classe discendente nella gerarchia di generalizzazione corrisponde a 1 RET.
Attributo semplice	Corrisponde a 1 DET.
Attributo complesso	Corrisponde a 1 RET.
Associazione con molteplicità max > 1	Aumenta la complessità di 1 RET.
Ereditarietà multipla	Ciascuna superclasse di una particolare classe corrisponde a 1 RET.
Gerarchia di classi	Ciascuna sottoclasse corrisponde a 1 RET.
Aggregazione	Aumenta la complessità di 1 RET.
Dipendenza tra classi	Non ha impatto nel conteggio dei function point.
External Input Pattern	Se tutti i messaggi di scrittura rivolti verso la Entity (= Entity Writing Message Sequence – EWMS) hanno argomenti, viene conteggiata una funzione transazionale EI.
External Output Pattern	Se i messaggi rivolti verso l'attore contengono tutti gli attributi recuperati mediante i messaggi di lettura rivolti verso la Entity (= Entity Reading Message Sequence – ERMS), viene conteggiata una funzione transazionale EQ. Se invece ne contengono alcuni ma non tutti, viene conteggiata una funzione transazionale EO.
External Input Pattern + External Output Pattern	Il numero di argomenti del messaggio che sono attributi di oggetti Entity corrisponde al conteggio dei DET; d'altra parte, il numero di oggetti Entity che partecipano allo scambio di messaggi corrisponde al conteggio dei FTR.

Rosso = use case diagram

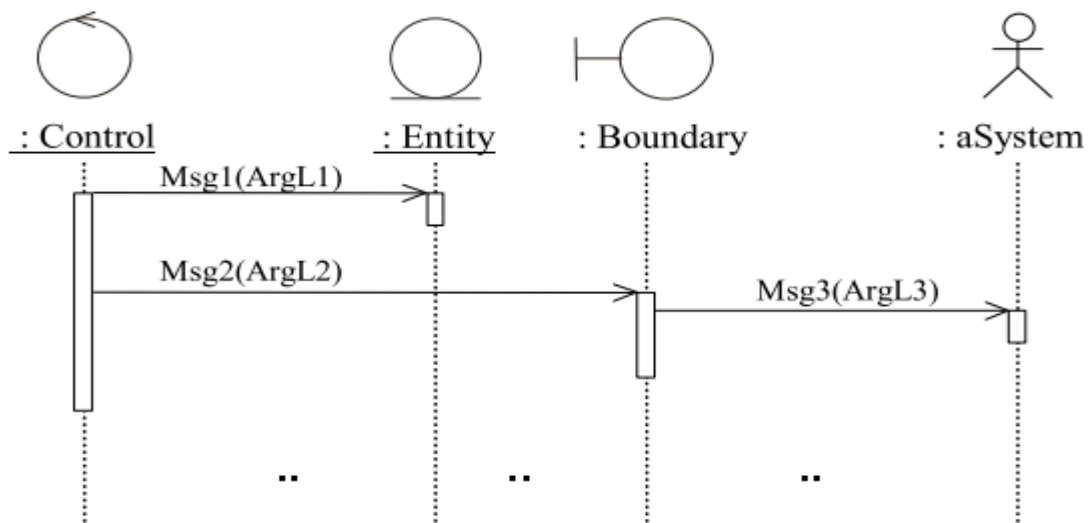
Blu = class diagram

Verde = sequence diagram

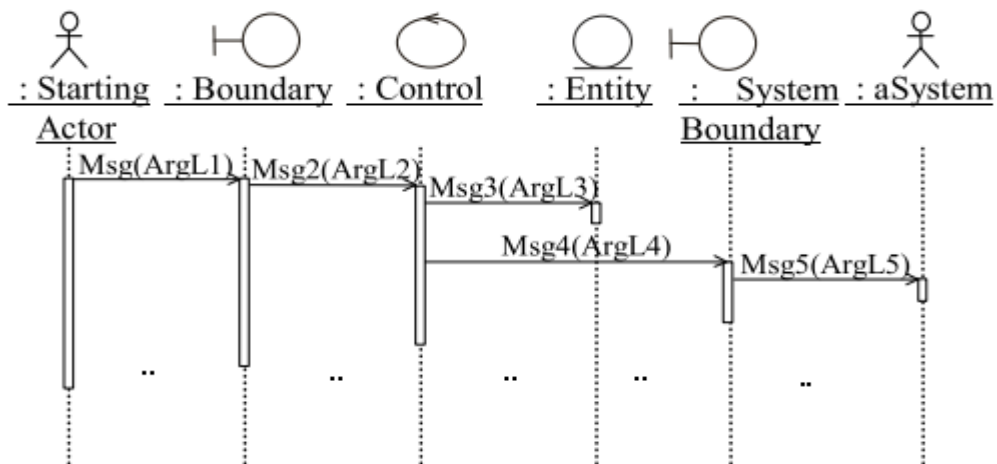
### External Input Pattern:



External Output Pattern:



Composed Pattern:



# MODELLAZIONE DEL COSTO DI UN PRODOTTO SOFTWARE

## CoCoMo I

CoCoMo I è un **modello predittivo di misura** che serve a stimare il costo di un software a seguito delle seguenti assunzioni:

- Processo software: il modello di riferimento è quello a cascata con le fasi di requisiti, progettazione, implementazione e testing attraversate sequenzialmente.
- Requisiti: sono essenzialmente considerati stabili.
- Qualità del progetto architetturale: il progetto è svolto da un piccolo numero di persone molto capaci e con una profonda conoscenza del livello applicativo.
- Progetto di dettaglio e costruzione in parallelo: il progetto di dettaglio dell'applicazione, il suo sviluppo e i suoi test unitari sono compiuti col massimo parallelismo possibile dai diversi progettisti / programmatori.
- Documentazione: è scritta in modo incrementale durante l'intero corso del progetto.
- Gestione: vi è una struttura di gestione del progetto.

**CoCoMo I** è da intendersi un modello statico e analitico: statico per quanto concerne le variabili di ingresso e di uscita; analitico in quanto può essere applicato anche a porzioni di un progetto.

Esistono tre diversi livelli di CoCoMo I che si differenziano per la precisione con cui vengono effettuate le stime: Basic CoCoMo, Intermediate CoCoMo e Advanced CoCoMo.

### Basic CoCoMo:

La stima del costo di basa sulla dimensione (Size S) del software espressa in migliaia di linee di codice (KLOC) e sulla sua complessità (Development mode). In particolare, le grandezze da stimare sono l'**effort (M) in mesi-persona** e il **tempo solare di sviluppo (T)**, e le formule da applicare sono le seguenti:

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

Complessità	$a_b$	$b_b$	$c_b$	$d_b$
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

### Intermediate CoCoMo:

Il valore di **M** va **moltiplicato per un certo C**, che è dato dal prodotto di 15 coefficienti correttivi (Cost drives) dati da:

- Il livello di **affidabilità** richiesto al software prodotto.
- La **stabilità** degli ambienti utilizzati per lo sviluppo e l'esecuzione dell'applicazione.
- L'**esperienza** media del team di sviluppo nel settore applicativo considerato.
- Il livello qualitativo delle tecniche di sviluppo utilizzate nel progetto, etc...

### Advanced CoCoMo:

A differenza dal livello intermedio, nel livello avanzato di CoCoMo i coefficienti correttivi variano da fase a fase.

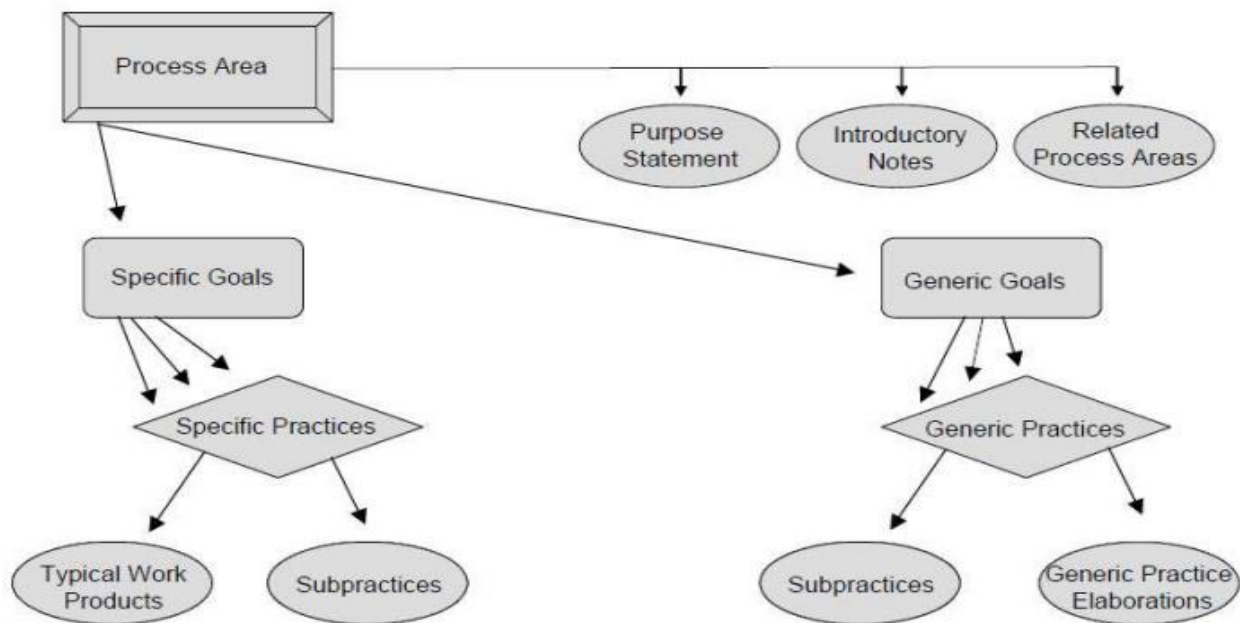
## MODELLI DI IMPROVEMENT

### CMMI

Il presupposto fondamentale del modello CMMI è il seguente: migliore è il processo per sviluppare e mantenere un sistema software, maggiore dovrebbe essere la qualità del sistema stesso.

CMMI, inoltre, si basa su un insieme definito di aree di processo (PA), che sono insieme di pratiche correlate che, se eseguite tutte insieme, concorrono al raggiungimento di obiettivi importanti per apportare miglioramenti in quell'area.

Un'area di processo si articola in obiettivi generici (comuni a più aree di processo) e obiettivi specifici (pensati esclusivamente per quell'area di processo). Ciascun obiettivo a sua volta può essere conseguito seguendo determinate pratiche.



Le aree di processo in CMMI sono in tutto 22 e sono suddivise in 4 categorie: gestione del progetto, gestione dei processi, ingegneria e supporto.

#### Project Management

- Project Planning
- Project Monitoring and Control
- Supplier Agreement Management
- Integrated Project Management(IPPD)
- Integrated Supplier Management (SS)
- Integrated Teaming (IPPD)
- Risk Management
- Quantitative Project Management

#### Support

- Configuration Management
- Process and Product Quality Assurance
- Measurement and Analysis
- Causal Analysis and Resolution
- Decision Analysis and Resolution
- Organizational Environment for Integration (IPPD)

#### Engineering

- Requirements Management
- Requirements Development
- Technical Solution
- Product Integration
- Verification
- Validation

#### Process Management

- Organizational Process Focus
- Organizational Process Definition
- Organizational Training
- Organizational Process Performance
- Organizational Innovation and Deployment

In CMMI sono previsti due approcci per migliorare le prestazioni dell'organizzazione: Staged Representation (SRn) e Continuous Representation (CRn).

#### Staged Representation:

È una rappresentazione che offre un modo sistematico e strutturato per affrontare il miglioramento del processo basato sul modello "una fase per volta": il miglioramento del processo porta a un upgrade parallelo e allineato di tutte le aree di processo.

I livelli di maturità previsti sono cinque:

- 1) **Initial:** l'organizzazione produce gli artefatti senza una gestione vera e propria: le scadenze non sempre vengono rispettate e si tende a gettare la spugna nei periodi di maggiore stress.
- 2) **Managed:** l'organizzazione gestisce in modo opportuno i progetti ed effettua una pianificazione del lavoro da svolgere. Il budget e le tempistiche vengono valutate in maniera adeguata. I processi vengono regolamentati in brevi documenti e si riesce a conseguire tutti gli obiettivi fissati anche nei periodi di maggiore stress.
- 3) **Defined:** i processi, le pianificazioni e le metodologie utilizzate all'interno dell'organizzazione sono ben definiti, sono ad hoc per ciascun progetto e sono riportati in documenti molto più dettagliati rispetto al caso "managed". Viene assunto un approccio proattivo anziché reattivo di fronte alle esigenze e alle emergenze.
- 4) **Quantitatively Managed:** i processi e i prodotti vengono caratterizzati mediante aspetti quantitativi, e gli obiettivi che si vogliono conseguire sono tutti misurabili. Ciò permette di verificare con precisione quanto bene viene gestito il lavoro e com'è la qualità dei processi e dei prodotti.
- 5) **Optimizing:** si punta a migliorare la qualità, la pianificazione e la gestione del lavoro in modo continuativo. Inoltre, è richiesto un continuo aggiornamento delle tecnologie, delle strategie, dei pattern e dei tool utilizzati all'interno dell'organizzazione, in modo tale da risultare sempre competitivi sul mercato.

#### Continuous Representation:

È una rappresentazione che offre un modo per migliorare le prestazioni su una o più aree di processo, eventualmente trattando le diverse aree di processo con ritmi differenti. Ci sono comunque alcune limitazioni a riguardo a causa delle dipendenze tra alcune aree di processo.

I livelli di maturità previsti sono sei:

- 1) **Incomplete:** le pratiche per conseguire gli obiettivi relativi a una data area di processo non vengono seguite o vengono seguite solo parzialmente.
- 2) **Performed:** le pratiche per conseguire gli obiettivi relativi a una data area di processo vengono seguite ma solo per conseguire gli obiettivi generici e non quelli specifici come qualità, costi e pianificazione. D'altronde, le prestazioni potrebbero non essere stabili.
- 3) **Managed:** si ha una gestione del processo che porta al conseguimento sia degli obiettivi generici che di quelli specifici.
- 4) **Defined:** il processo è ben definito e viene attuato sulla base degli standard dell'organizzazione.
- 5) **Quantitatively Managed:** il processo viene gestito con un supporto di tecniche statistiche o comunque quantitative. Gli obiettivi inoltre devono risultare tutti misurabili.
- 6) **Optimizing:** il processo è soggetto a un monitoraggio continuo, a una pianificazione continua e a un miglioramento continuo. Nel caso in cui si riscontrano problemi, devono essere apportate le dovute correzioni al processo il più velocemente possibile; le correzioni includono l'aggiornamento della documentazione del processo.

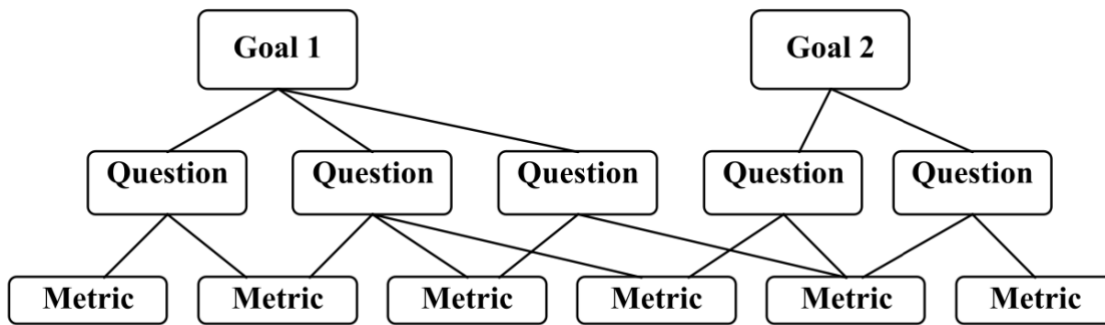
## **GQM**

Tale approccio si basa sul presupposto che per misurare in modo mirato un'organizzazione deve prima specificare gli obiettivi per sé stessa e i suoi progetti.

*Livello concettuale (GOAL):* un obiettivo è definito per un oggetto, per una serie di ragioni, rispetto a vari modelli di qualità, da vari punti di vista, relativi a un particolare ambiente. Gli oggetti di misura sono prodotti, processi, risorse.

*A livello operativo (Question)* si usano domande volte a caratterizzare la valutazione.

*A livello quantitativo (Metric)* ad ogni domanda lego dei dati utili ad una risposta.



# FINE TEORIA

### 1) Cosa significa “misurare”?

“Misurare” significa effettuare una misurazione, ovvero esercitare su un attributo di una particolare entità un processo che abbia lo scopo di ottenere un risultato di misurazione; il risultato di misurazione è un valore appartenente alla scala utilizzata che viene associato all’attributo a seguito dell’operazione di misurazione.

### 2) Cosa consente di fare la misurazione di un’entità?

Consente di:

- Caratterizzare, cioè estrarre delle informazioni da una proprietà (attributo) dell’entità.
- Comparare una proprietà dell’entità con l’analoga proprietà di un’entità simile.
- Controllare e monitorare l’entità.
- Valutare, cioè dare un valore a una proprietà dell’entità.
- Predire il valore di una proprietà dell’entità.

### 3) A cosa servono le scale?

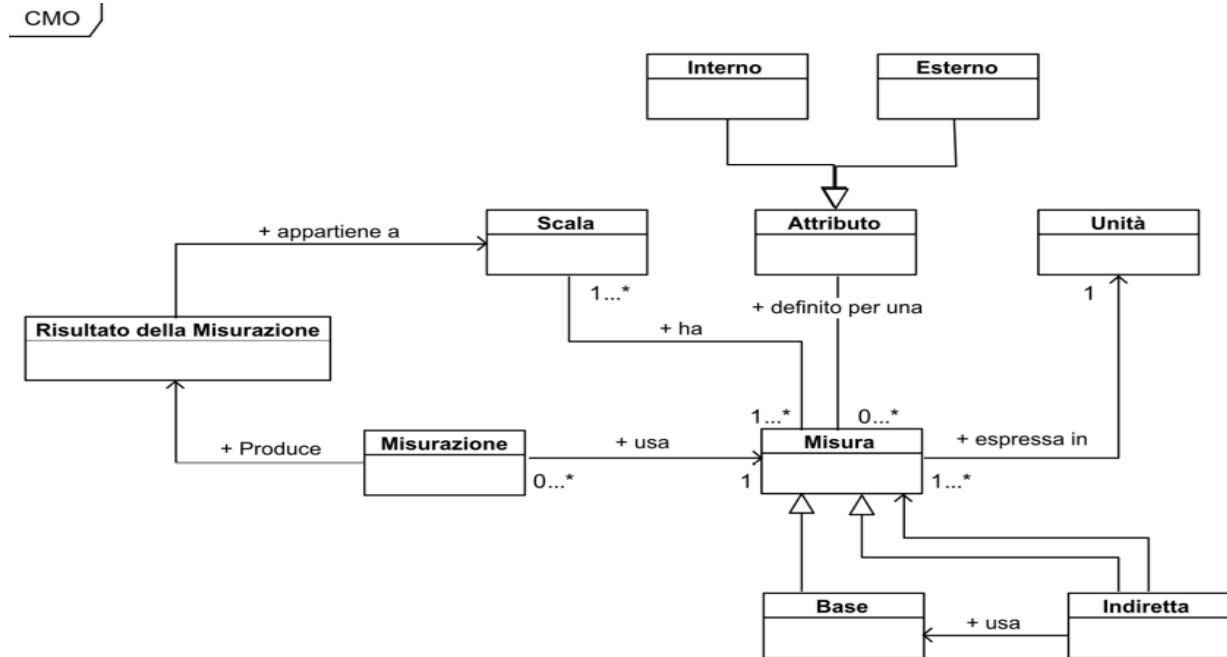
Le scale servono a definire il dominio dei valori che possono essere assegnati all’attributo dell’entità che si sta misurando; inoltre, forniscono un’indicazione univoca e ben precisa su cosa indica ciascun valore che può essere associato all’attributo stesso.

### 4) Puoi descrivere nel dettaglio la CMO?

La CMO (Classic Measurement Ontology) è un **modello di rappresentazione formale** secondo cui:

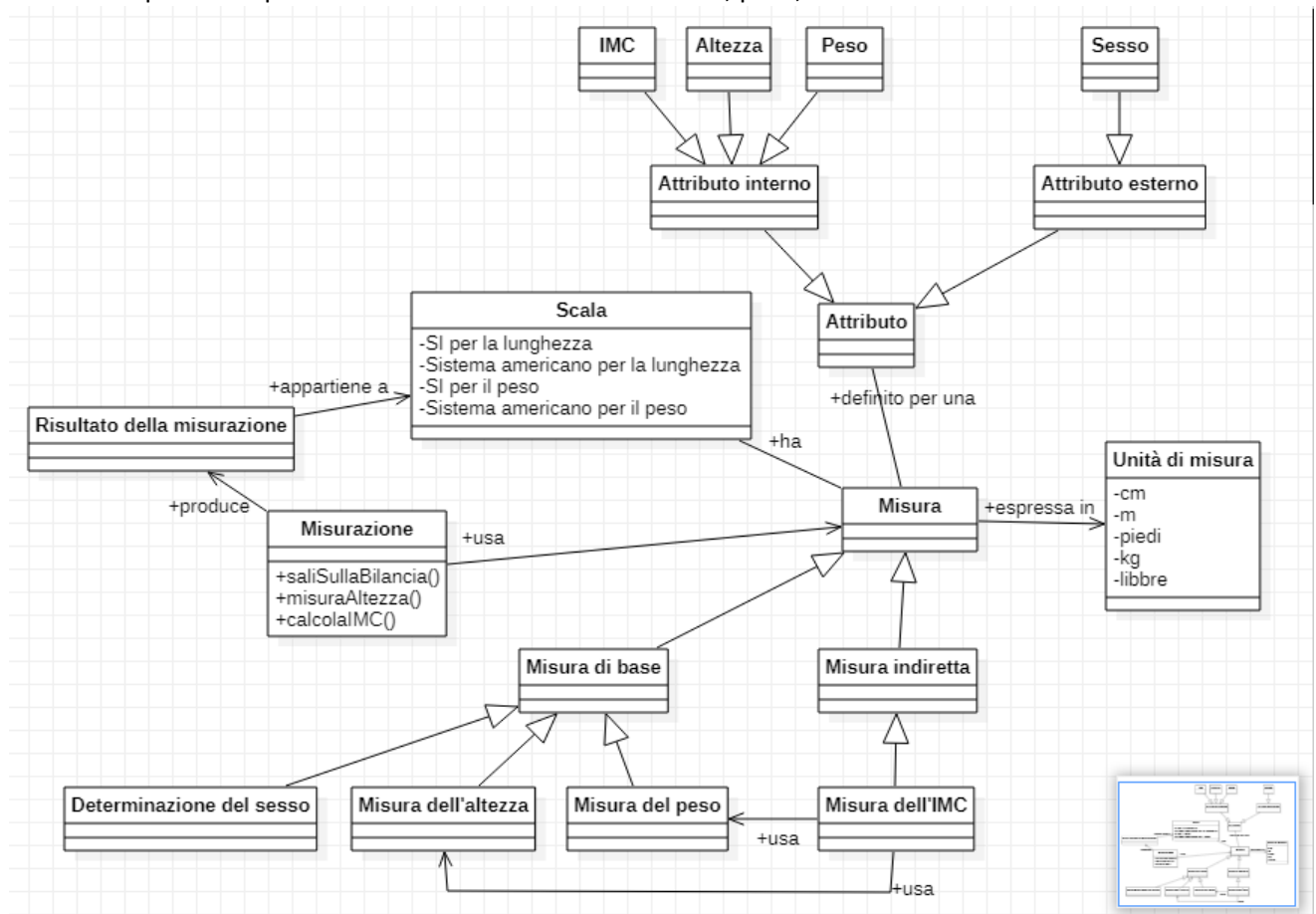
- Un attributo può essere definito per una misura, ovvero può essere associato tramite quest’ultima a un valore appartenente a una determinata scala.
- Un attributo può essere esterno (se è osservabile direttamente a partire da un’entità differente) oppure interno (se è osservabile solo prendendo l’entità isolatamente).
- Ciascuna misura è espressa in un’unità di misura.
- Una misura può essere di base oppure indiretta; quest’ultima usa una o più misure di base.
- Una misurazione fa uso di una misura per produrre il risultato della misurazione.
- Il risultato della misurazione appartiene a una determinata scala, e non è altro che il valore che viene associato a un attributo di un’entità a seguito del processo di misurazione.
- Ciascuna misura ha una o più scale; è possibile effettuare delle trasformazioni da una scala all’altra (dette somiglianze) mediante delle operazioni matematiche.

In conclusione, la CMO può essere rappresentata mediante il seguente diagramma UML:



5) Creare un esempio reale istanziando in uno specifico dominio applicativo la CMO. Descrivere in dettaglio il diagramma UML risultante e il dominio di partenza.

Dominio di partenza: persona di cui si vuole conoscere sesso, peso, altezza e indice IMD.



6) Definire le metriche funzionali e non funzionali.

Le metriche funzionali sono metriche del software che misurano aspetti riguardanti i requisiti funzionali del



sistema (= ciò che il sistema dovrebbe fare). Le metriche non funzionali, invece, sono metriche del software che misurano tutto ciò che riguarda i requisiti non funzionali (= come il sistema dovrebbe offrire le proprie funzionalità).

### 7) Cosa misurano le metriche funzionali?

Le metriche funzionali misurano la quantità e l'effort richiesto per implementare i requisiti funzionali del sistema.

### 8) Cosa misurano le metriche non funzionali?

Le metriche non funzionali misurano aspetti del sistema software concernenti la sua qualità e le sue caratteristiche non funzionali come la disponibilità, l'affidabilità, l'usabilità, la manutenibilità, la sicurezza e così via.

### 9) Cosa sono i function point?

I function point sono l'unità di misura che quantifica la funzionalità aziendale di un sistema (senza però tener conto dei requisiti non funzionali).

### 10) In cosa consiste il calcolo dei function point?

Il conteggio dei function point restituisce una funzione di ILF (file logici interni), EIF (file di interfaccia esterna), EI (input esterni), EO (output esterni), EQ (richieste esterne), RET (record element type), DET (data element type) e FTR (file type referenced). In particolare, per effettuare il calcolo dei function point bisogna seguire i seguenti passaggi (di cui gli ultimi due in realtà sono facoltativi):

- I) Definire il tipo di conteggio.
- II) Definire il limite del conteggio.
- III) Individuare gli EP (processi elementari) del sistema.
- IV) Identificare gli EP unici.
- V) Misurare le funzioni dati del sistema.
- VI) Misurare le funzioni di transizione del sistema.
- VII) Calcolare il conteggio dei function point non aggiustato.
- VIII) Calcolare il coefficiente di aggiustamento (VAF).
- IX) Calcolare il conteggio dei function point aggiustato.

### 11) È possibile automatizzare il calcolo dei FP?

Sì purché non si utilizzi lo use case diagram come supporto. Non si differenziano gli utenti! (es: stampante vs attore)

### 12) Come può aiutare la documentazione UML di un sistema software a calcolare i FP?

La documentazione UML è d'aiuto mediante lo **use case diagram**, il **class diagram** e il **sequence diagram**. In particolare:

Elemento	Calcolo da effettuare
Attore umano	È sorgente / destinazione di funzioni transazionali.
Attore non umano	È sorgente / destinazione di funzioni transazionali.
Altro tipo di attore (e.g. stampante, timer)	Deve essere scartato dall'analisi dei function point.
Associazione tra caso d'uso e attore	È candidato per alcune funzioni transazionali (e.g. EI, EO, EQ).
Caso d'uso di estensione o di inclusione	È candidato per alcune funzioni transazionali (e.g. EI, EO, EQ).
Classe Entity	È candidata per ILF o EIF.
Classe non Entity	Viene sempre scartata (per cui non rientra nei casi esposti dopo).
Classe astratta	Ciascuna classe discendente nella gerarchia di generalizzazione corrisponde a 1 RET.
Attributo semplice	Corrisponde a 1 DET.

Attributo complesso	Corrisponde a 1 RET.
Associazione con molteplicità max > 1	Aumenta la complessità di 1 RET.
Ereditarietà multipla	Ciascuna superclasse di una particolare classe corrisponde a 1 RET.
Gerarchia di classi	Ciascuna sottoclasse corrisponde a 1 RET.
Aggregazione	Aumenta la complessità di 1 RET.
Dipendenza tra classi	Non ha impatto nel conteggio dei function point.
External Input Pattern	Se tutti i messaggi rivolti verso la Entity hanno argomenti, viene conteggiata una funzione transazionale EI.
External Output Pattern	Se gli argomenti di tutti i messaggi includono alcuni ma non tutti gli attributi letti dalle Entity, viene conteggiata una funzione transazionale EO; se invece includono tutti gli attributi, viene conteggiata una funzione transazionale EQ; negli altri casi il pattern viene scartato.
External Input Pattern + External Output Pattern	Il numero di argomenti del messaggio che sono attributi di oggetti Entity corrisponde al conteggio dei DET; d'altra parte, il numero di oggetti Entity che partecipano allo scambio di messaggi corrisponde al conteggio dei FTR.

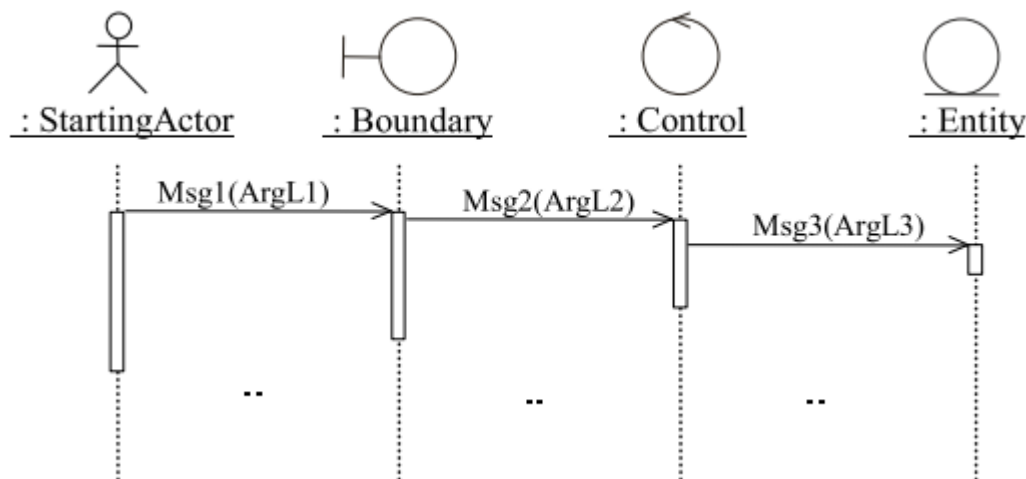
### 13) Quale documentazione risulta essere la migliore per l'individuazione delle funzioni transazionali?

La documentazione data dai sequence diagram, in cui effettivamente gli EI, EO, EQ sono molto precisi e facili da individuare.

### 14) Cosa sono i pattern nell'ambito UML-FP? Come si usano?

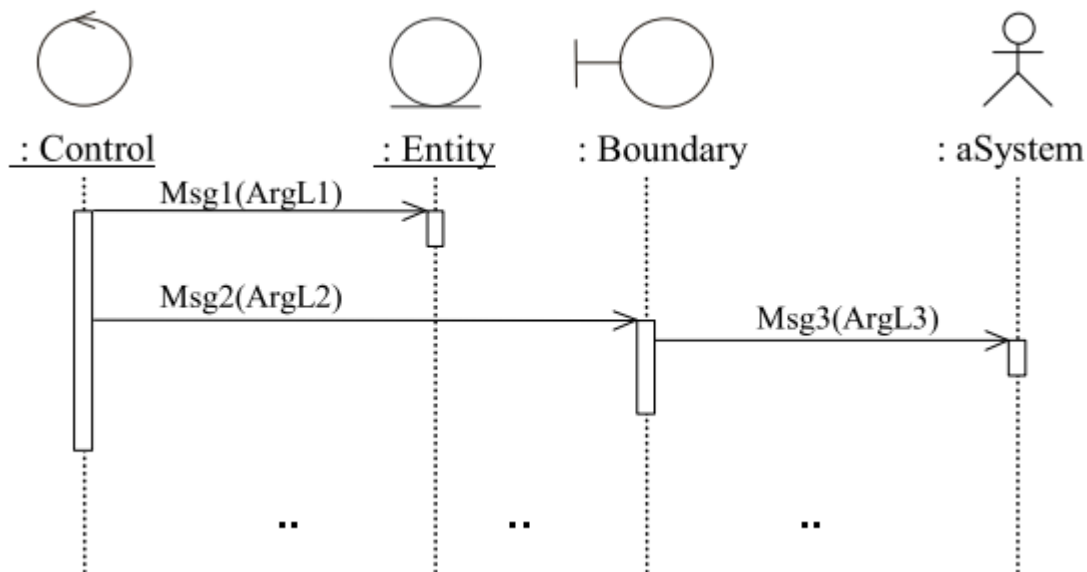
I pattern nell'ambito UML-FP sono sequenze di messaggi ben precise che si possono trovare all'interno dei sequence diagram e sono:

- External Input Pattern, identificato da un System Directed Messages Sequence (SDMS):



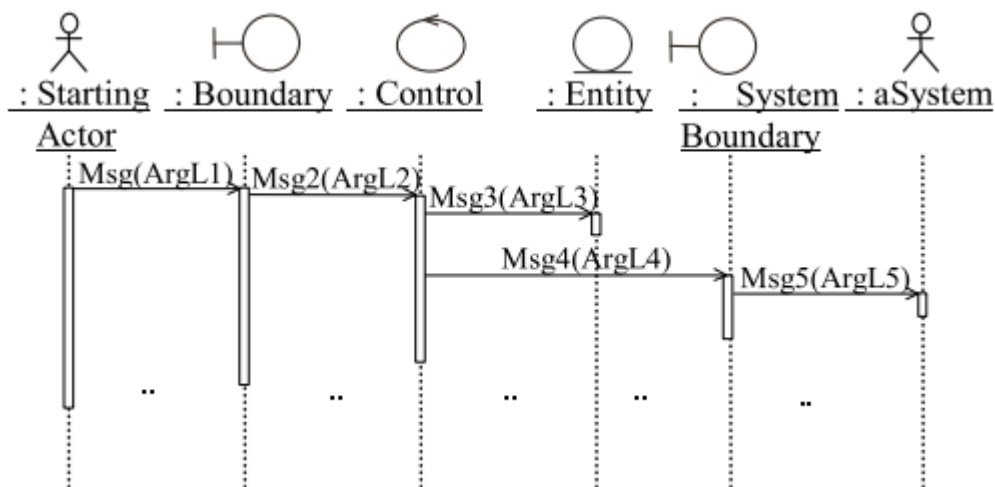
Qui, se tutti i messaggi di scrittura rivolti verso la Entity (= Entity Writing Message Sequence – EWMS) hanno argomenti, viene conteggiata una funzione transazionale EI.

- External Output Pattern, identificato da un Actor Directed Messages Sequence (ADMS):



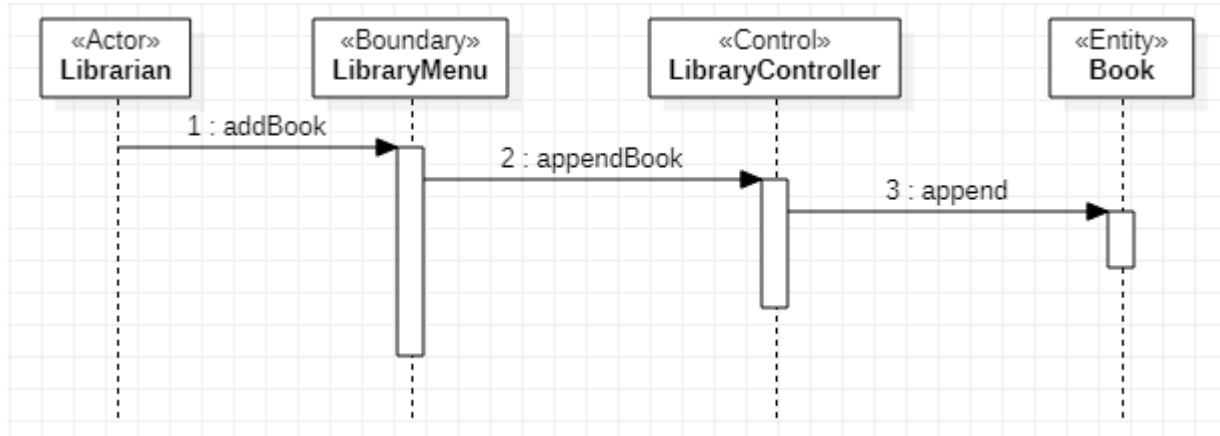
Qui, se i messaggi rivolti verso l'attore contengono tutti gli attributi recuperati mediante i messaggi di lettura rivolti verso la Entity (= Entity Reading Message Sequence – ERMS), viene conteggiata una funzione transazionale EQ. Se invece ne contengono alcuni ma non tutti, viene conteggiata una funzione transazionale EO.

- Composed Pattern, in cui gli External Input Pattern ed External Output Pattern vanno estratti e valutati separatamente:



15) Si è in grado di fare esempi, anche semplici, di utilizzo di elementi in UML per l'individuazione dei Processi Elementari?

Processo Elementare “Aggiungi libro”:



### 16) Esistono metodologie e/o modelli per stimare il costo di un prodotto software?

Certamente: i più importanti sono i modelli parametrici, tra cui spiccano CoCoMo (Constructive Cost Model), PRICE-S (Parametric Review of Information for Costing and Evaluation – Software), SEER-SEM (Software Evaluation and Estimation of Resources – Software Estimating Model) e SLIM (Software Lifecycle Management).

### 17) In cosa consiste il modello CoCoMo I? Quali sono le sue caratteristiche principali?

CoCoMo I è un modello che serve a stimare il costo di un software a seguito delle seguenti assunzioni:

- Processo software: il modello di riferimento è quello a cascata con le fasi di requisiti, progettazione, implementazione e testing attraversate sequenzialmente.
- Requisiti: sono essenzialmente considerati stabili.
- Qualità del progetto architetturale: il progetto è svolto da un piccolo numero di persone molto capaci e con una profonda conoscenza del livello applicativo.
- Progetto di dettaglio e costruzione in parallelo: il progetto di dettaglio dell'applicazione, il suo sviluppo e i suoi test unitari sono compiuti col massimo parallelismo possibile dai diversi progettisti / programmatori.
- Documentazione: è scritta in modo incrementale durante l'intero corso del progetto.
- Gestione: vi è una struttura di gestione del progetto.

CoCoMo I è da intendersi un modello statico e analitico: statico per quanto concerne le variabili di ingresso e di uscita; analitico in quanto può essere applicato anche a porzioni di un progetto.

Esistono tre diversi livelli di CoCoMo I che si differenziano per la precisione con cui vengono effettuate le stime: Basic CoCoMo, Intermediate CoCoMo e Advanced CoCoMo.

#### Basic CoCoMo:

La stima del costo di basa sulla dimensione (Size S) del software espressa in migliaia di linee di codice (KLOC) e sulla sua complessità (Development mode). In particolare, le grandezze da stimare sono l'effort

(M) in mesi-persona e il tempo solare di sviluppo (T), e le formule da applicare sono le seguenti:

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

Complessità	$a_b$	$b_b$	$c_b$	$d_b$
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

#### Intermediate CoCoMo:

Il valore di M va moltiplicato per un certo C, che è dato dal prodotto di 15 coefficienti correttivi (Cost drives) dati da:

- Il livello di affidabilità richiesto al software prodotto.
- La stabilità degli ambienti utilizzati per lo sviluppo e l'esecuzione dell'applicazione.
- L'esperienza media del team di sviluppo nel settore applicativo considerato.
- Il livello qualitativo delle tecniche di sviluppo utilizzate nel progetto.
- E così via.

#### Advanced CoCoMo:

A differenza dal livello intermedio, nel livello avanzato di CoCoMo i coefficienti correttivi variano da fase a fase.

#### **18) Gli attributi e i cost drivers del modello di CoCoMo sono indipendenti gli uni dagli altri? Sono interdipendenti?**

I cost drivers del modello di CoCoMo non sono del tutto indipendenti tra loro: ad esempio, se il livello di affidabilità richiesto al software prodotto è molto elevato, sarebbe impensabile assumere un team di sviluppo non qualificato e con poca esperienza.

#### **19) Quali sono le dimensioni critiche per un processo organizzativo nella vista SEI?**

Nella vista SEI (Software Engineering Institute) gli aspetti più critici sono quelli legati alla maturità delle organizzazioni. In particolare gli errori da evitare sono:

- Processo improvvisato durante i progetti.
- Processi approvati ignorati.
- Approccio reattivo anziché proattivo.
- Budget e programma non realistici.
- Qualità sacrificata per stare nei tempi.
- Nessuna misura oggettiva della qualità.

#### **20) Qual è il presupposto fondamentale alla base degli approcci CMMI?**

Il presupposto fondamentale è il seguente: migliore è il processo per sviluppare e mantenere un sistema software, maggiore dovrebbe essere la qualità del sistema stesso.

#### **21) Cos'è l'area di processo PA?**

Un'area di processo è un insieme di pratiche correlate che, se eseguite tutte insieme, concorrono al raggiungimento di obiettivi importanti per apportare miglioramenti in quell'area.

#### **22) Quante aree di processo CMMI prende in considerazione?**

CMMI prende in considerazione 22 aree di processo suddivise in quattro categorie (gestione del progetto, gestione dei processi, ingegneria e supporto).

**23) Elenca tre o più PA.**

- Causal Analysis and Resolution (CAR)
- Configuration Management (CM)
- Decision Analysis and Resolution (DAR)
- Integrated Project Management + Integrated Process & Product Development (IPM+IPPD)
- Measurement and Analysis (MA)
- Organizational Innovation and Deployment (OID)
- Organizational Process Definition + Integrated Process & Product Development (OPD+IPPD)
- Organizational Process Focus (OPF)
- Organizational Process Performance (OPP)
- Organizational Training (OT)
- Product Integration (PI)
- Project Monitoring and Control (PMC)
- Project Planning (PP)
- Process and Product Quality Assurance (PPQA)
- Quantitative Project Management (QPM)
- Requirements Development (RD)
- Requirements Management (REQM)
- Risk Management (RSKM)
- Supplier Agreement Management (SAM)
- Technical Solution (TS)
- Validation (VAL)
- Verification (VER)

**24) Quanti e quali livelli di maturità sono considerati dalla Rappresentazione Staged in CMMI?**

Nella Rappresentazione Staged in CMMI sono considerati cinque livelli di maturità:

- Initial (1)
- Managed (2)
- Defined (3)
- Quantitatively Managed (4)
- Optimizing (5)

**25) Spiegare le relazioni tra PA, pratiche, obiettivi generici e obiettivi specifici in CMMI.**

Un'area di processo si articola in obiettivi generici (comuni a più aree di processo) e obiettivi specifici (pensati esclusivamente per quell'area di processo). Ciascun obiettivo a sua volta può essere conseguito seguendo determinate pratiche.

**26) Elenca il focus di ogni livello di maturità CMMI (Rappresentazione Staged).**

- Initial: l'organizzazione produce gli artefatti senza una gestione vera e propria: le scadenze non sempre vengono rispettate e si tende a gettare la spugna nei periodi di maggiore stress.
- Managed: l'organizzazione gestisce in modo opportuno i progetti ed effettua una pianificazione del lavoro da svolgere. Il budget e le tempistiche vengono valutate in maniera adeguata. I processi vengono regolamentati in brevi documenti e si riesce a conseguire tutti gli obiettivi fissati anche nei periodi di maggiore stress.
- Defined: i processi, le pianificazioni e le metodologie utilizzate all'interno dell'organizzazione sono ben definiti, sono ad hoc per ciascun progetto e sono riportati in documenti molto più dettagliati rispetto al caso "managed". Viene assunto un approccio proattivo anziché reattivo di fronte alle esigenze e alle emergenze.
- Quantitatively Managed: i processi e i prodotti vengono caratterizzati mediante aspetti quantitativi, e gli obiettivi che si vogliono conseguire sono tutti misurabili. Ciò permette di verificare con precisione quanto

bene viene gestito il lavoro e com'è la qualità dei processi e dei prodotti.

- Optimizing: si punta a migliorare la qualità, la pianificazione e la gestione del lavoro in modo continuativo. Inoltre, è richiesto un continuo aggiornamento delle tecnologie, delle strategie, dei pattern e dei tool utilizzati all'interno dell'organizzazione, in modo tale da risultare sempre competitivi sul mercato.

## **27) Quante e quali capacità sono considerate nella Rappresentazione Continua in CMMI?**

Nella Rappresentazione Continua in CMMI sono considerati sei livelli di maturità:

- Incomplete (0)
- Performed (1)
- Managed (2)
- Defined (3)
- Quantitatively Managed (4)
- Optimizing (5)

## **28) Spiegare ciascun livello CMMI nella Rappresentazione Continua.**

- Incomplete: le pratiche per conseguire gli obiettivi relativi a una data area di processo non vengono seguite o vengono seguite solo parzialmente.

- Performed: le pratiche per conseguire gli obiettivi relativi a una data area di processo vengono seguite ma solo per conseguire gli obiettivi generici e non quelli specifici come qualità, costi e pianificazione.

D'altronde, le prestazioni potrebbero non essere stabili.

- Managed: si ha una gestione del processo che porta al conseguimento sia degli obiettivi generici che di quelli specifici.

- Defined: il processo è ben definito e viene attuato sulla base degli standard dell'organizzazione.

- Quantitatively Managed: il processo viene gestito con un supporto di tecniche statistiche o comunque quantitative. Gli obiettivi inoltre devono risultare tutti misurabili.

- Optimizing: il processo è soggetto a un monitoraggio continuo, a una pianificazione continua e a un miglioramento continuo. Nel caso in cui si riscontrano problemi, devono essere apportate le dovute correzioni al processo il più velocemente possibile; le correzioni includono l'aggiornamento della documentazione del processo.