

**Progetto di sviluppo,  
Struttura Organizzativa,  
Qualità nello sviluppo FOSS**

## **Uno sforzo TEMPORANEO intrapreso per creare un Prodotto (o un Servizio) UNICO**

### **□ Temporaneo**

- Indica che ha un ben definito inizio e una ben definita fine
  - Termina quando si sono raggiunti gli obiettivi oppure è chiaro che NON si possono raggiungere gli obiettivi
- NON indica necessariamente una temporaneità del prodotto realizzato durante il progetto

### **□ Unico**

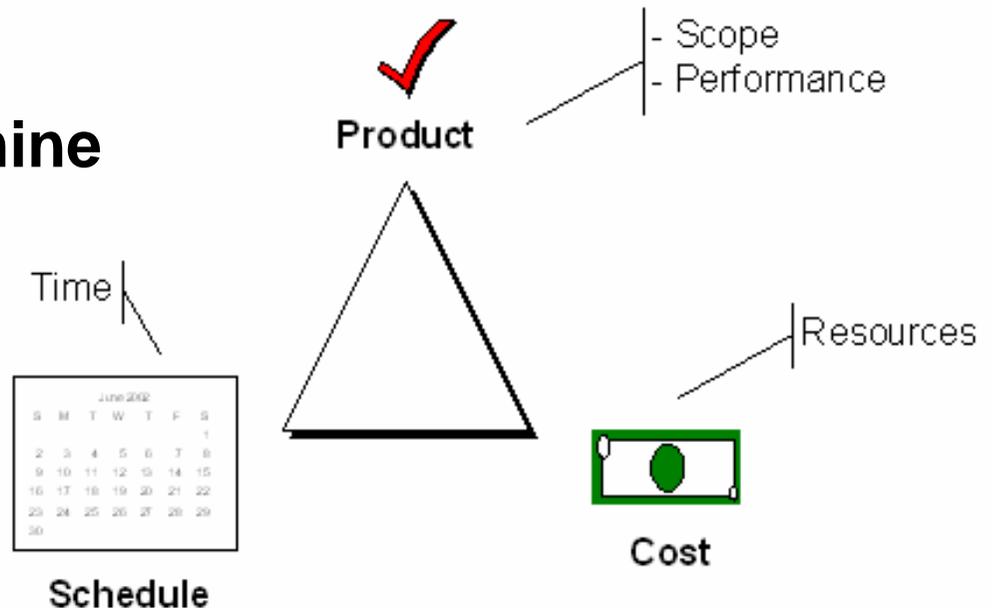
- Indica che il prodotto (o servizio) è diverso per qualche aspetto distintivo dagli altri prodotti (o servizi)

## □ Obiettivi strategici

- **Motivazioni strategiche che portano ad attivare un progetto**
- **Risultati attesi come eredità del progetto**
  - Rispetto a questi obiettivi si valuta l'investimento giustificando l'assegnazione delle risorse e l'assunzione di eventuali rischi

## □ Obiettivi specifici

- **Risultati attesi al termine del progetto**
  - Qualità
  - Tempo
  - Budget



- ❑ **I progetti di sviluppo SW presentano alcune peculiarità specifiche che ne determinano**
  - Una elevata complessità
  - Una difficile pianificazione e conduzione
  
- ❑ **Per questo motivo è fondamentale una attenta gestione dei progetti SW applicando un *mix* di competenze legate a**
  - Project Management
  - Ingegneria del Software

**Il Focus del CapoProgetto è sul Processo**

**Il risultato di entrambi è il**

**PRODOTTO**

**Il Focus del Gruppo di Lavoro è sul Prodotto**

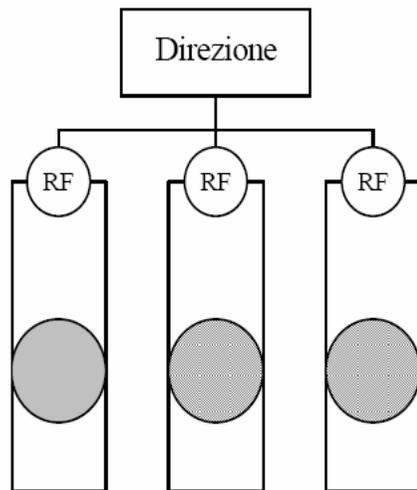
- ❑ **Pianificazione del progetto:** definire gli obiettivi, le strategie e i piani secondo i quali il progetto verrà condotto e sviluppato
- ❑ **Organizzazione del progetto:** definire la struttura organizzativa del gruppo di lavoro a cui verrà affidato lo svolgimento del progetto e l'assegnazione delle diverse responsabilità funzionali
- ❑ **Staffing:** identificare, reperire, formare e valutare le risorse umane necessarie allo svolgimento del progetto
- ❑ **Direzione** comprende tutte le attività legate alla conduzione del progetto e, in particolare, il coordinamento delle attività, il decision-making e la supervisione del personale
- ❑ **Controllo del progetto:** verificare che il progetto si svolga secondo i piani formulati durante la fase di pianificazione

- Il **Project Management** è l'applicazione di conoscenze, abilità, strumenti e tecniche alle attività del progetto per raggiungerne gli obiettivi
  
- Il **lavoro** può essere riassunto nelle seguenti attività
  - Definizione di obiettivi, tempi, costi, rischi e qualità
  - Gestire tutte le persone e organizzazioni coinvolte nel progetto (**stakeholder**) con le loro specifiche necessità (**esigenze esplicite**) e aspettative (**esigenze implicite**)
  - Gestire il progetto in tutti i suoi processi: inizio, pianificazione, esecuzione, controllo e chiusura

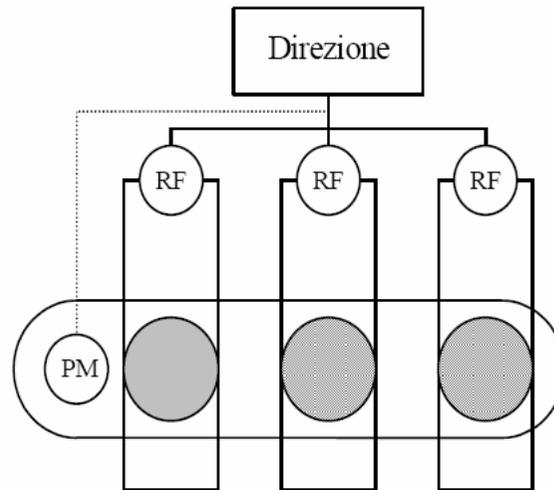
- ❑ Sono gli individui e le organizzazioni attivamente coinvolti nel progetto o i cui interessi possono essere influenzati positivamente o negativamente dall'esecuzione/successo del progetto
  - **Utilizzatore** – colui che usufruisce del prodotto, può coincidere o meno con il cliente
  - **Cliente** – Colui che sostiene l'investimento di un progetto esterno
  - **Committente** – colui che bandisce la richiesta di offerta ed è responsabile di fronte al Cliente del progetto per tutta la sua durata (funge da interfaccia per il Contractor)
  - **Contractor** – Azienda che stipula il contratto per la realizzazione del progetto
  - **Fornitore**: fornitore di servizi, prodotti, componenti o materie prime
- ❑ Il problema è gestire le diverse esigenze e aspettative (a volte conflittuali fra loro)
  - Normalmente si devono risolvere a favore del cliente

- ❑ I progetti coinvolgono una parte di una realtà più complessa la cui organizzazione influisce sull'organizzazione stessa del progetto

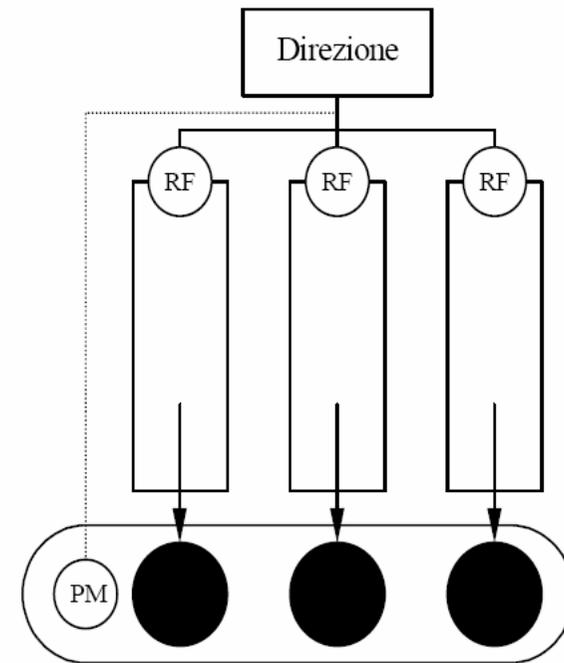
Struttura Funzionale



Struttura a Matrice



Task Force



Legenda:

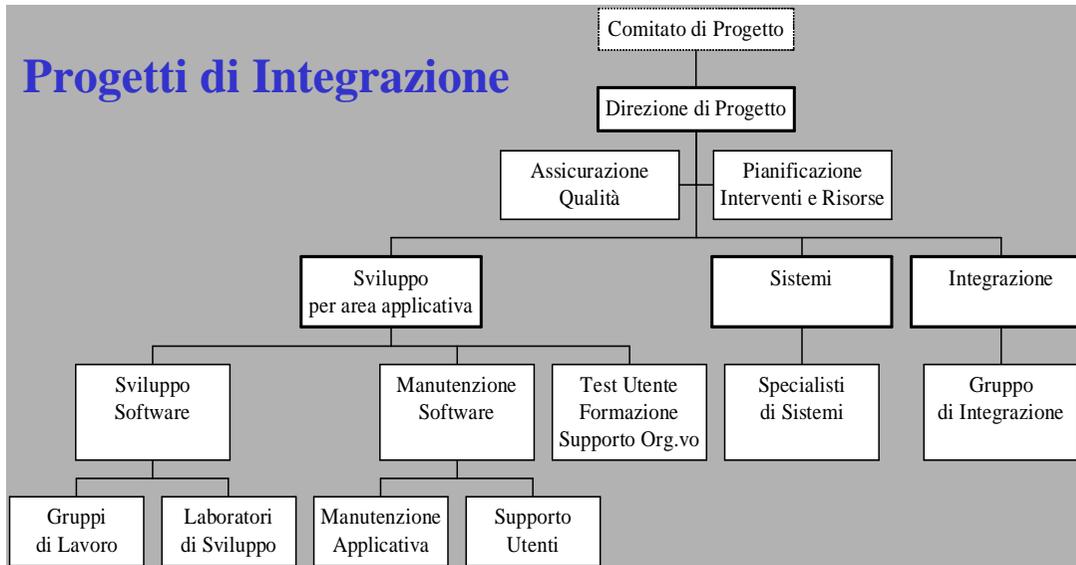
RF Responsabile Funzionale

PM Project Manager

● Team a tempo parziale

● Team a tempo pieno

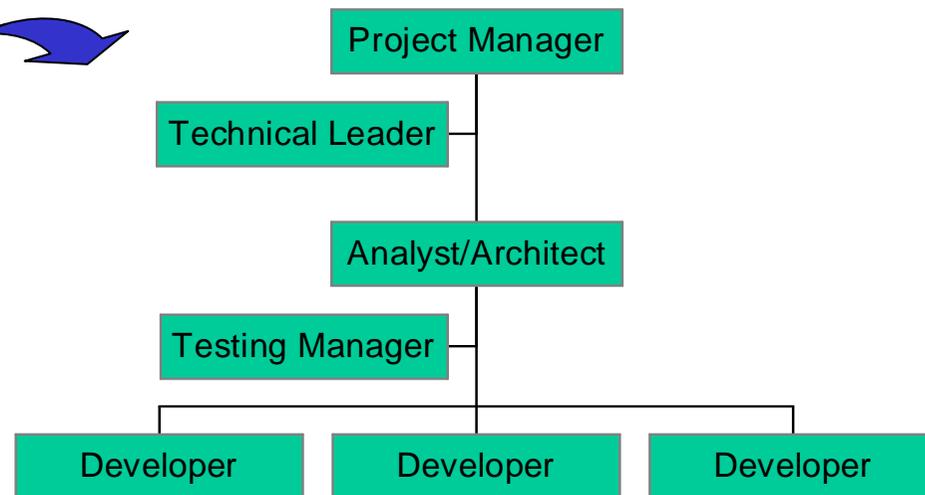
Fonte: MIP

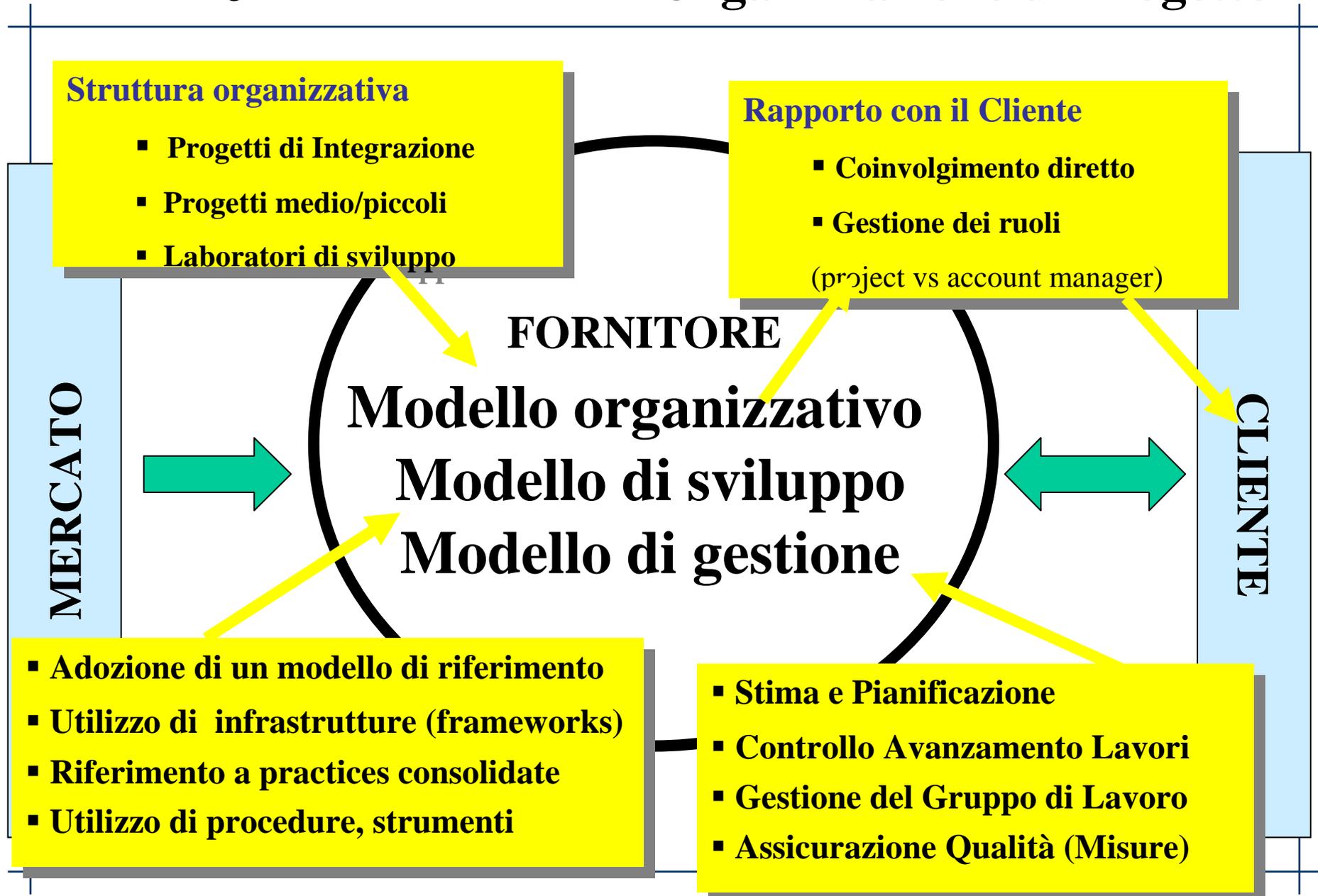


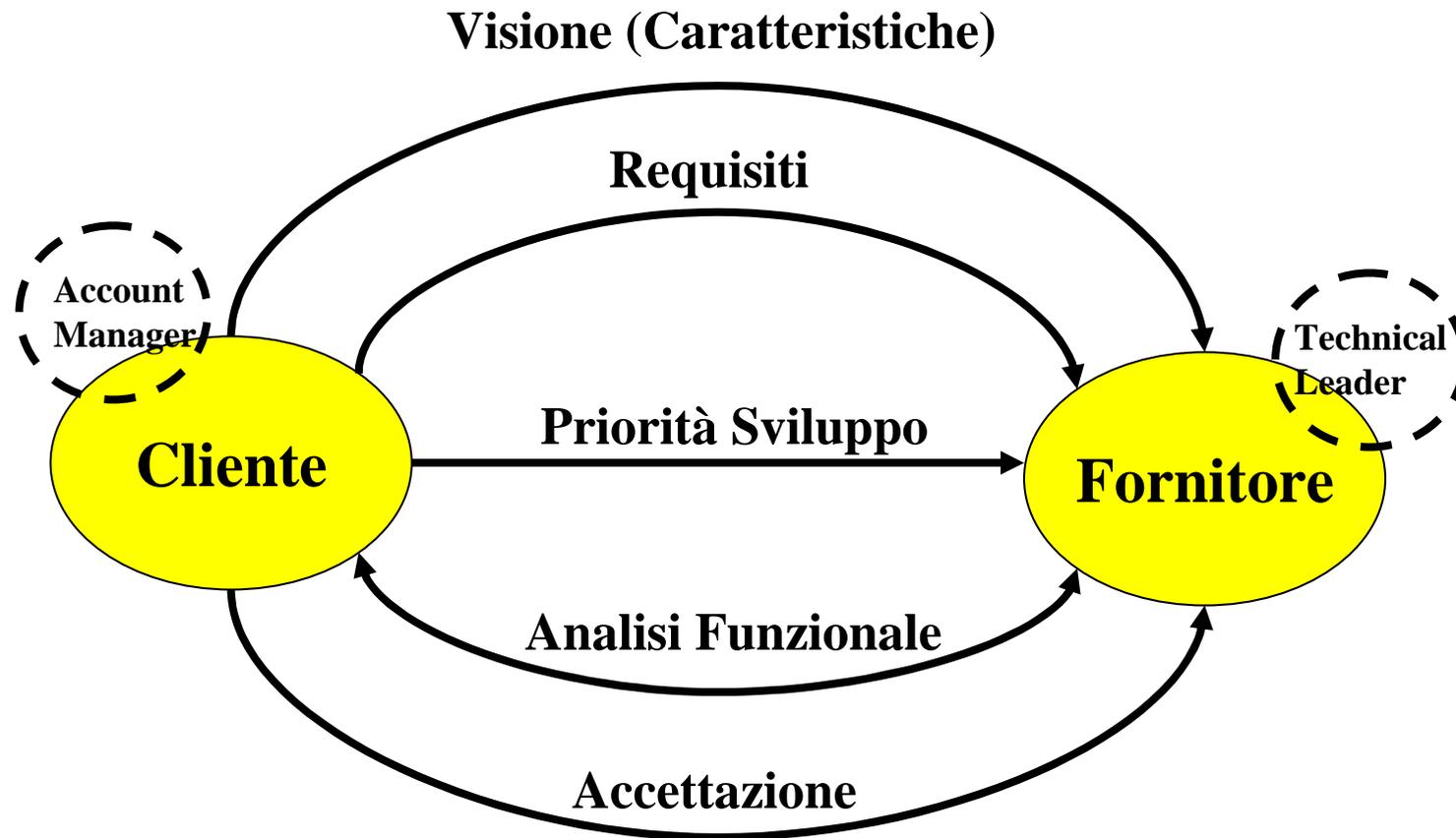
## Progetti Medio/Piccoli

### Laboratori di sviluppo

- Competenze specifiche
- Collaboration/Knowledge
- Rete di supporto

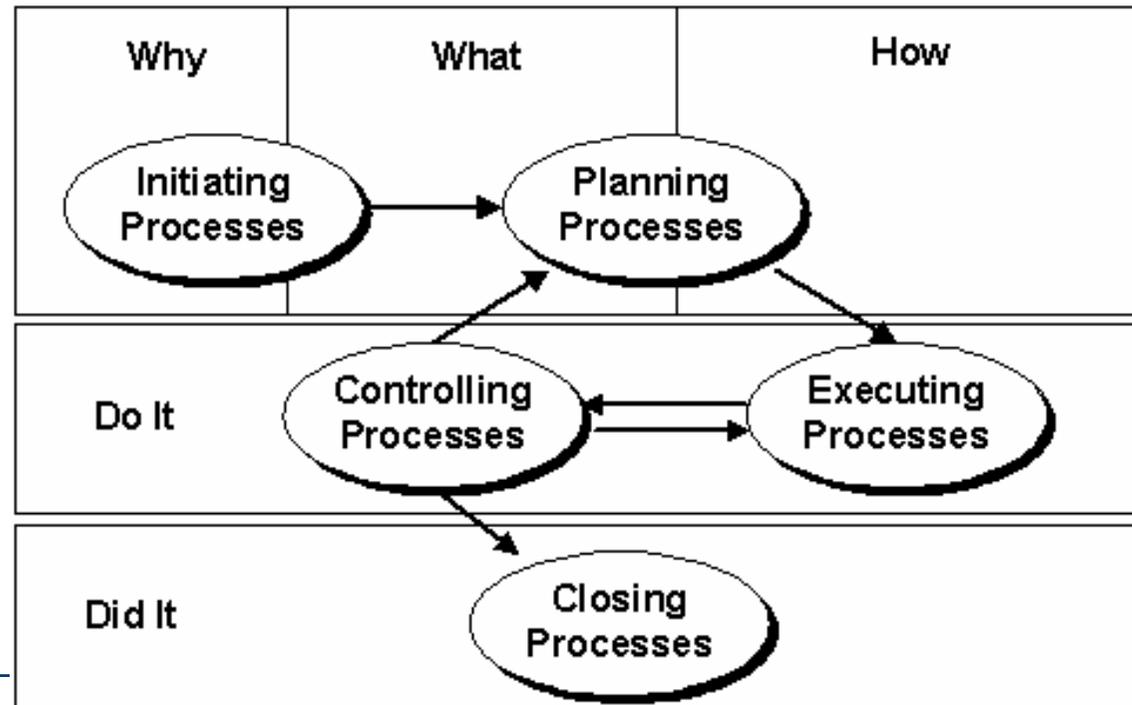






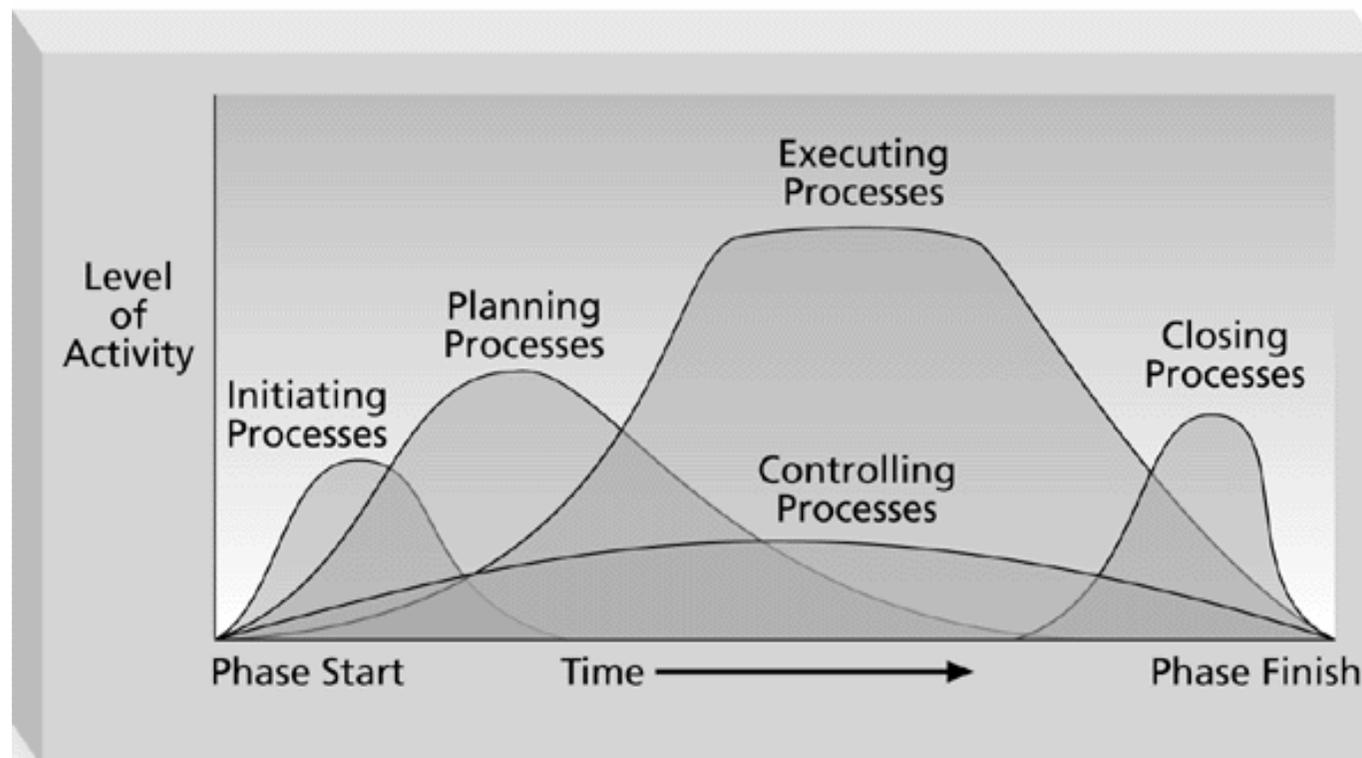
## PMI – Project Management Institute

- ❑ Why - Obiettivi
- ❑ What - Requisiti
- ❑ How – Specifiche di progetto, Piano di sviluppo, Ciclo di vita
- ❑ Do - Esecuzione
- ❑ Done – Review di fine progetto



Futrell, Shafer, “Quality Software Project Management”

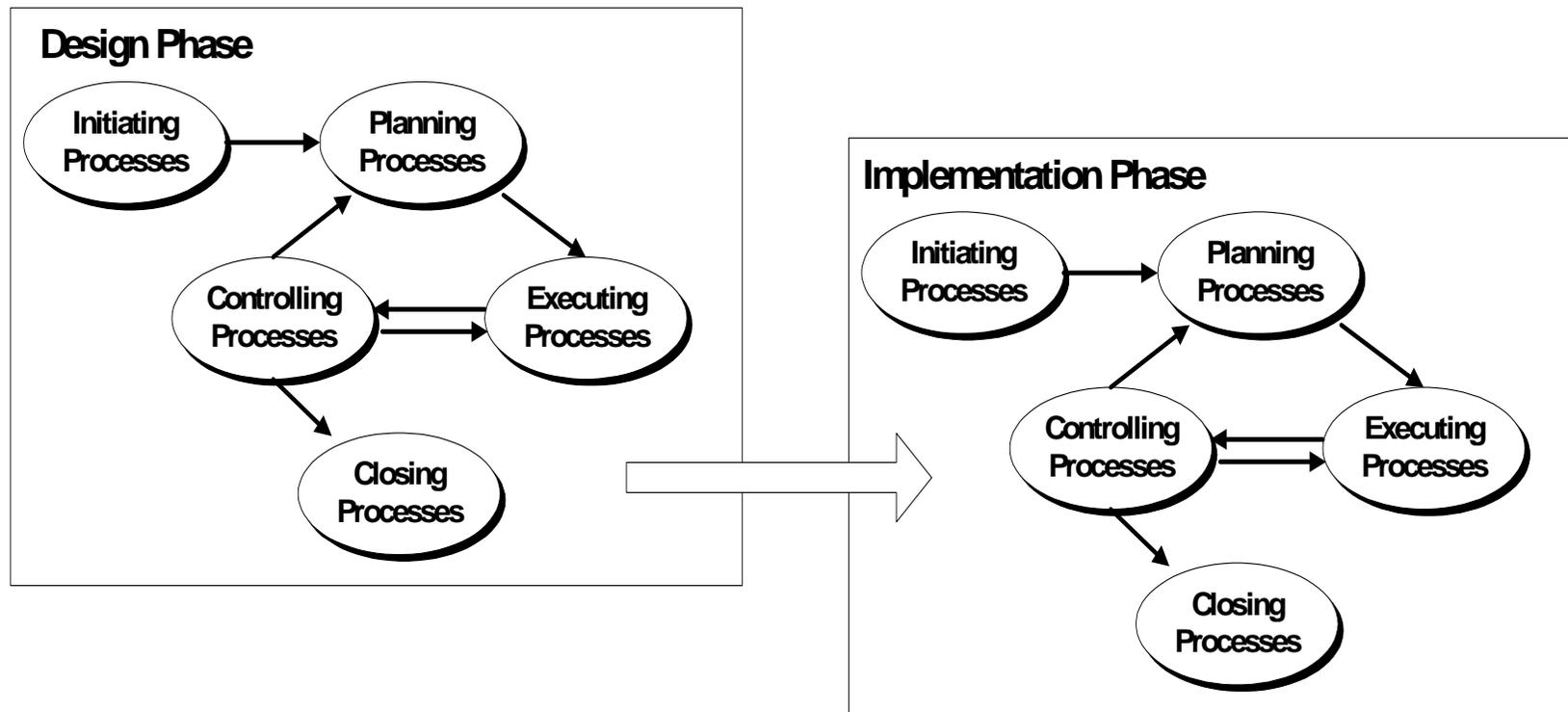
I processi di PM non sono rigidamente sequenziali, ma si sovrappongono nel tempo



Columbia University - Q7503 Principles of Software Project Management

<http://www.columbia.edu/~jm2217/>

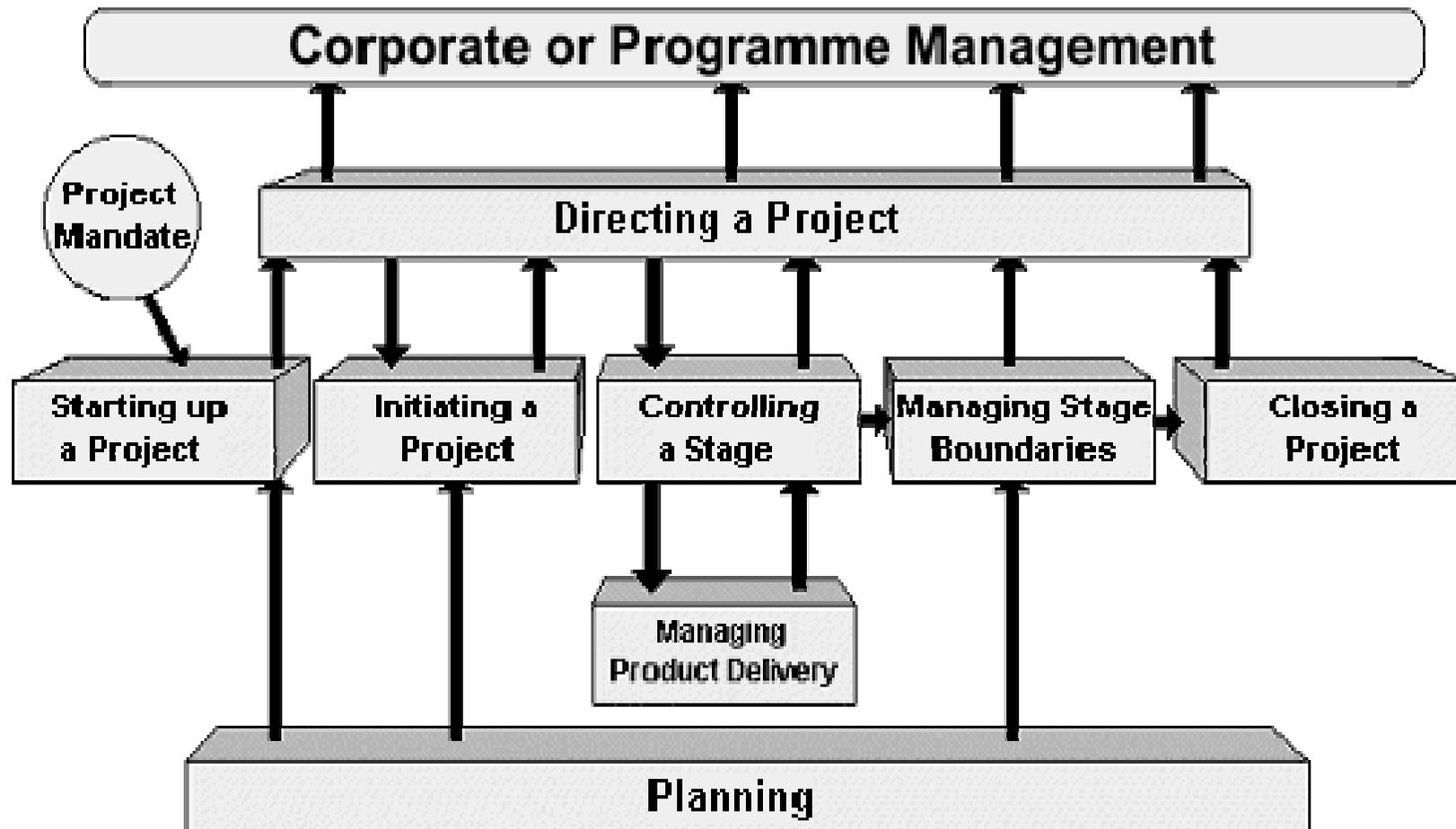
Intervengono in ogni fase del ciclo di vita del progetto

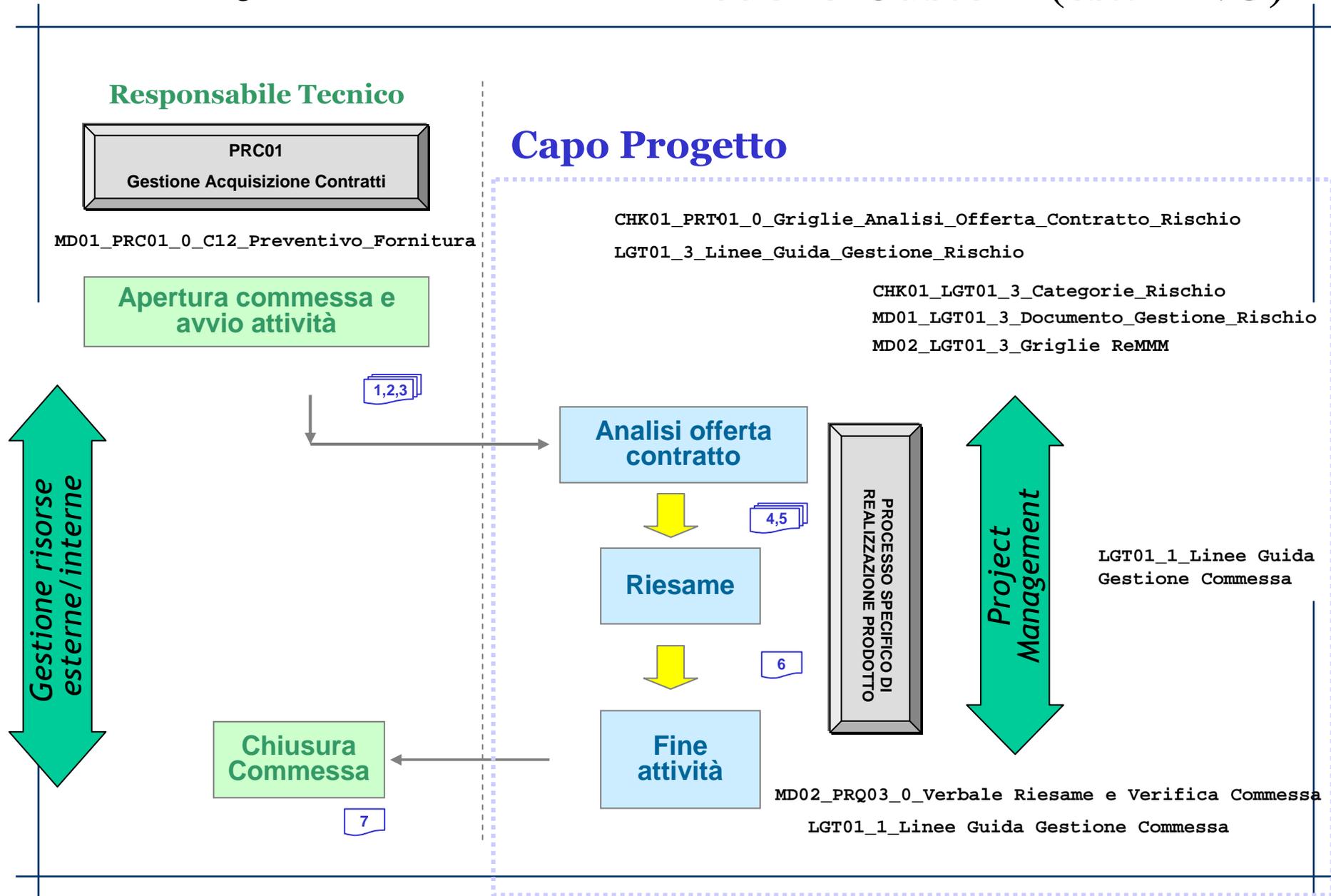


Columbia University - Q7503 Principles of Software Project Management

<http://www.columbia.edu/~jm2217/>

## PRojects In Controlled Environment





**Gestire il rischio di un progetto significa occuparsi  
attivamente del suo successo**

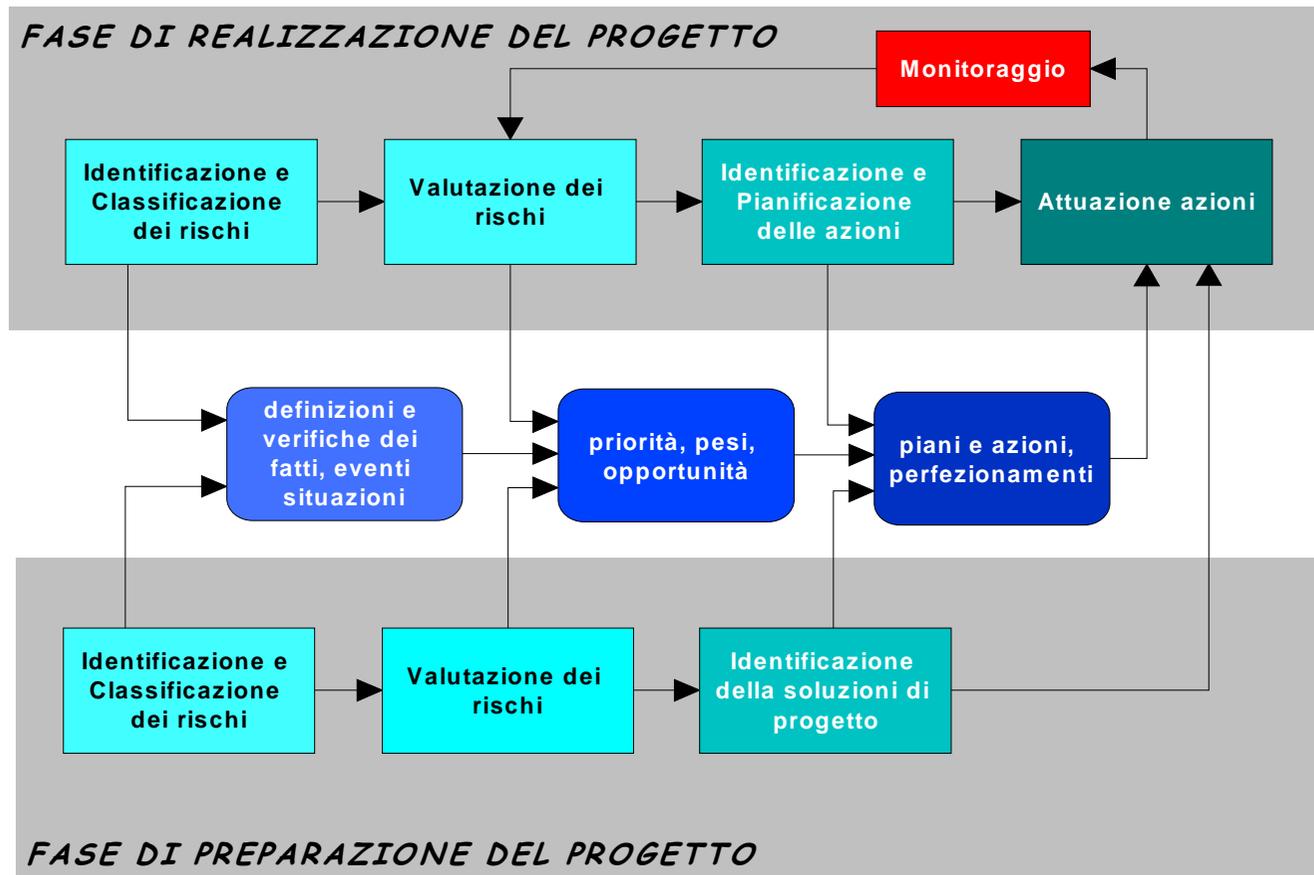
*(R. Meli)*

**Il rischio è connaturato ad ogni processo di sviluppo:**

- è legato ad eventi futuri
- è legato al cambiamento
- è collegato alle scelte (area di indeterminatezza)

**La gestione dei rischi è essenziale nei processi di  
sviluppo evolutivi/iterativi:**

- controllo delle iterazioni



**Il problema di “stimare i costi di sviluppo software” è critico per le organizzazioni che devono conoscere nel modo più accurato possibile la dimensione e l’impegno di realizzazione prima di procedere all’avvio di un progetto, momento in cui le informazioni utilizzate per la stima si possono basare su requisiti incompleti o errati.**

**La stima dello sviluppo software deve fornire due informazioni per decidere se affrontare lo sviluppo di un progetto:**

- la dimensione del costo di sviluppo***
- la dimensione del tempo necessario***

**attraverso la:**

- scomposizione del problema***
- individuazione del ciclo di sviluppo***
- stima delle dimensioni del software da sviluppare.***

## Metriche di stima

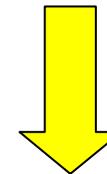
- ❑ Dimensionali interne  
*Source Lines of Code (SLOC)*
- ❑ Dimensionali esterne  
*(funzionali)*  
*Function Points (FP)*  
*Objects Points, Web Objects*

## Metodi di stima

- ❑ Metodi diretti (giudizio degli esperti, stime per analogia)
- ❑ Metodo derivati (algoritmici o parametrici)
- ❑ Stime per decomposizione
- ❑ Stime process oriented

### Metrica FP:

- External Inputs (EI)
- External Outputs (EO)
- External Inquiries (EQ)
- Internal file (ILF)
- External Interfaces (EIF)

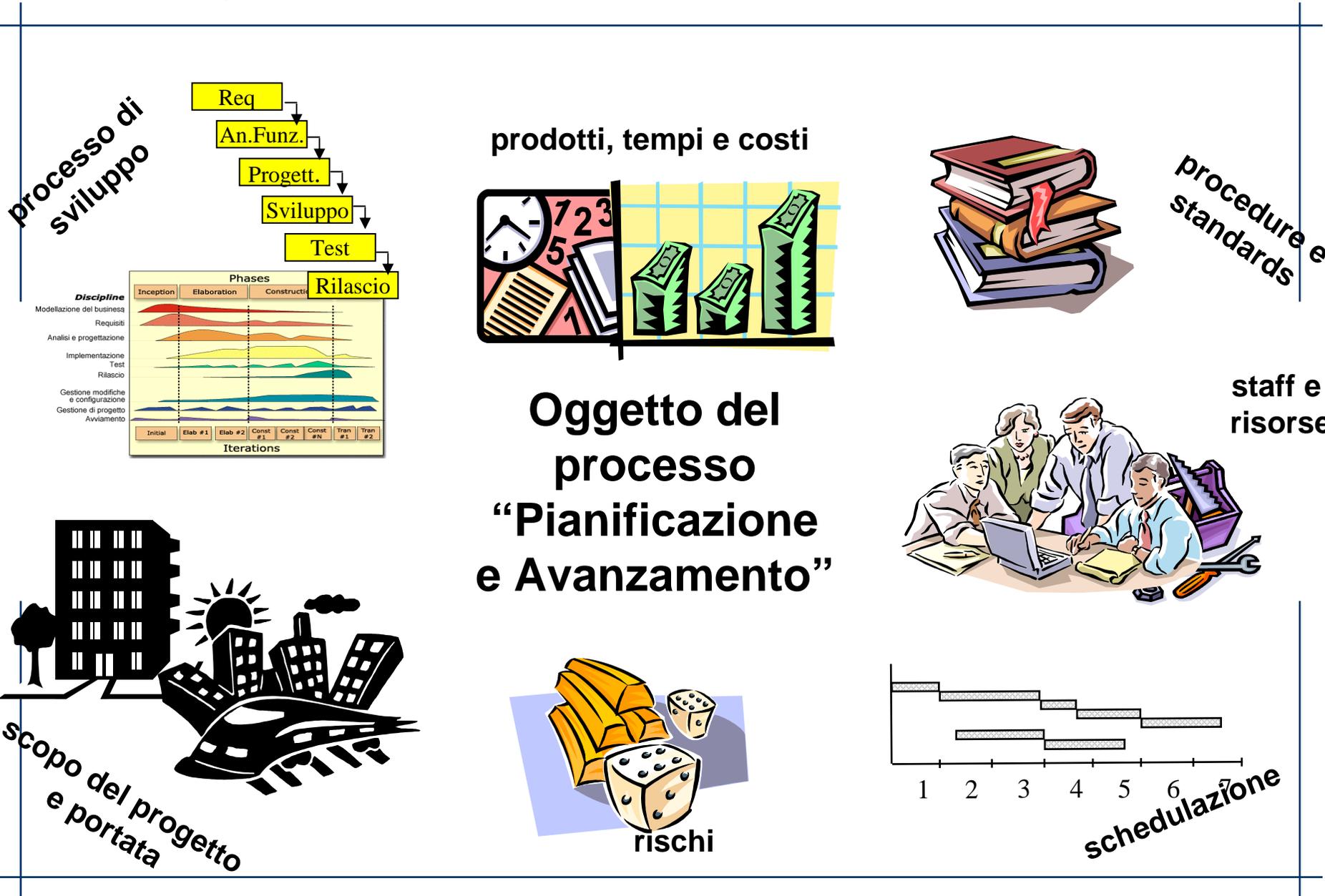


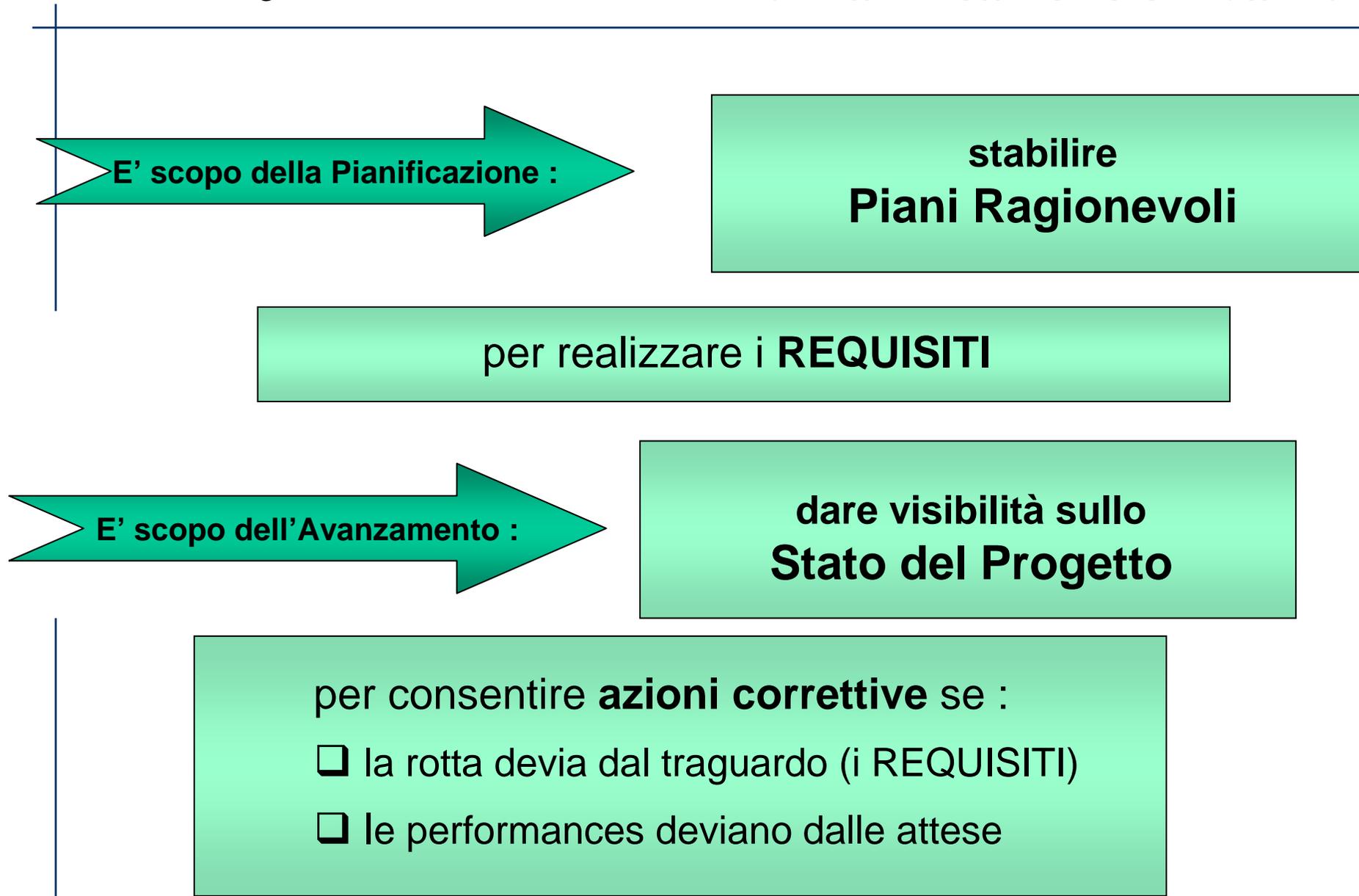
### Approccio tradizionale

- Giudizio degli esperti
- Function Point Analysis
- COCOMO

### Approccio evolutivo

- COCOMO II
- WEBMO (J. Reifer)
- EwebMO (Engineering)





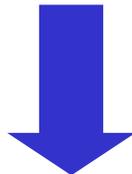
- Definizione degli obiettivi e ambito del progetto**
- Definizione delle attività e della loro collocazione temporale**
- Definizione della stima dei costi e budget**
- Definizione del piano di gestione dei rischi**

**Definizione/scelta  
del processo  
(ciclo di vita)**

**Pianificazione  
attività**

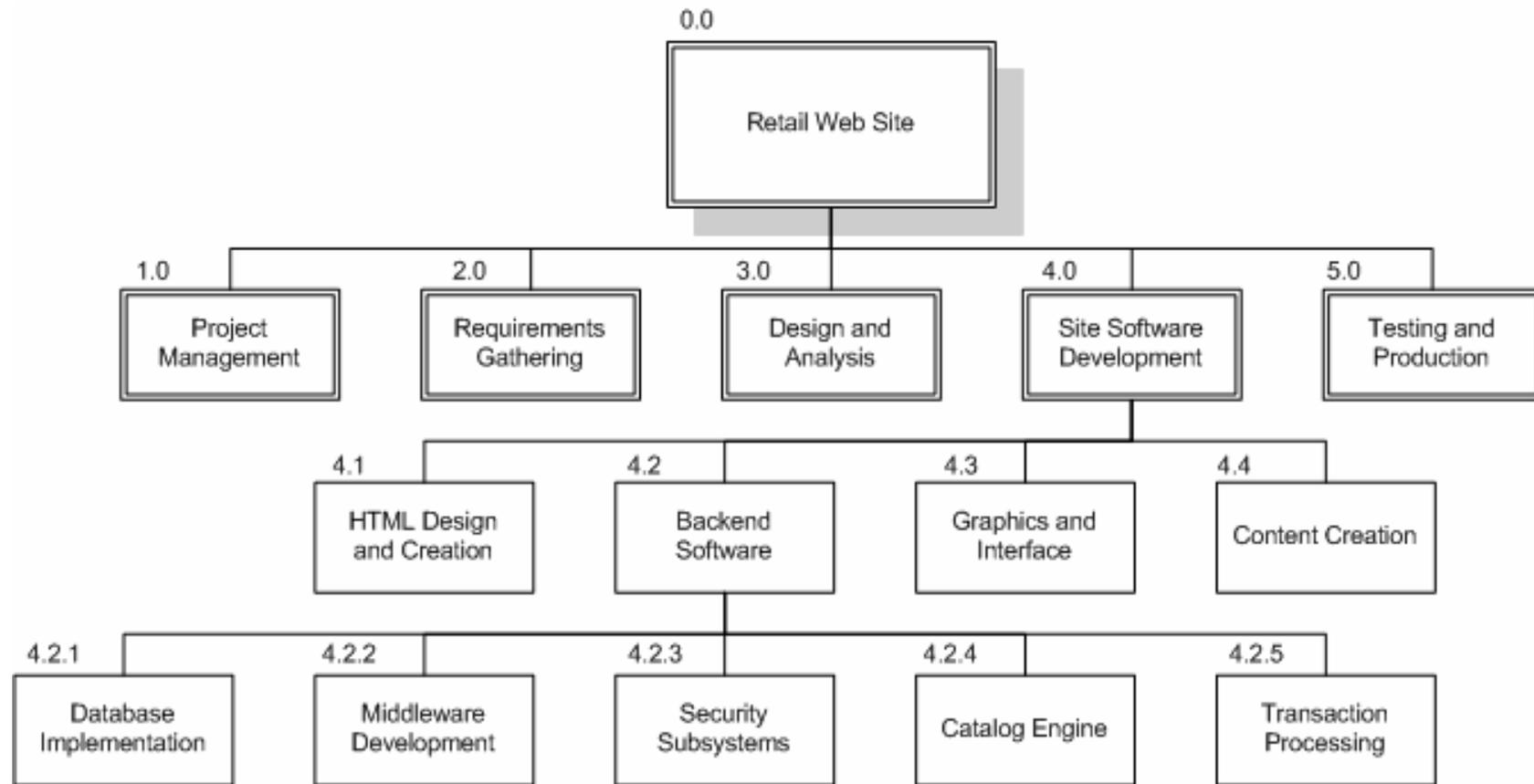
**Stime**

**Rischi**



- Stesura del piano di progetto**

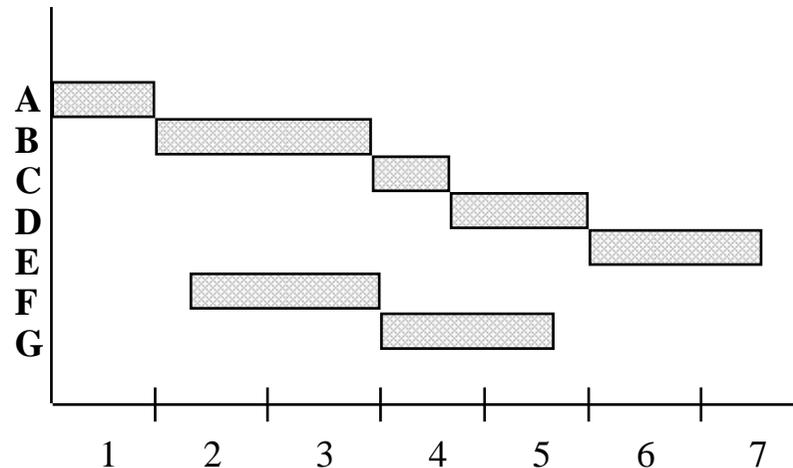
- La **Work Breakdown Structure** è lo strumento per la scomposizione del progetto nelle sue componenti
  - Parte dall'output della definizione di progetto (avvio) e identifica le attività alla base di tutta la pianificazione successiva
  
- E' il punto di partenza per la pianificazione in quanto identifica tutti i task di un progetto
  - Spesso viene definita "task list"
  
- Può essere rappresentata in forma grafica o testuale
  - Grafica → più immediata
  - Testuale → più pratica da usare



Columbia University - Q7503 Principles of Software Project Management

<http://www.columbia.edu/~jm2217/>

- ❑ Diagramma a barre che permette una visualizzazione efficace dell'allocazione temporale delle attività
- ❑ Tutte le attività sono distribuite sul diagramma facendo riferimento ad un unico asse temporale
- ❑ Possono essere rappresentati anche i legami di precedenza (frecche che collegano le attività)
- ❑ Strumento di comunicazione ad alto livello
  - Piano generale di progetto (fasi/macroattività)



## **CCM: Change&Configuration Management**

□ Procedure e tecniche per identificare univocamente, documentare le caratteristiche, controllare e documentare i cambiamenti di un componente documentale o software

## **SCM: Software Configuration Management**

□ Individua practices di project management che permettono la distribuzione accurata e riproducibile del software

## **VCS: Version Control System**

□ Individua il sistema (tool) utilizzato per implementare il SCM: gestisce il repository dove vengono immagazzinate le varie versioni di codice realizzato in sviluppo e/o manutenzione

- Ispezioni sul processo**
- Ispezioni sul prodotto**
- Raccolta di misure (processo e prodotto)**
- Analisi dati**
- Ciclo del Miglioramento**

## **BAZAAR STYLE (The Cathedral and the Bazaar)**

- ❑ Diversi rilasci, piccoli ed incrementali**
- ❑ Molti sviluppatori inviano patches e nuove proposte**
- ❑ Per il rilascio è sufficiente la compilazione e il funzionamento del sw (compile and run), successivamente la release viene testata in modo distribuito**

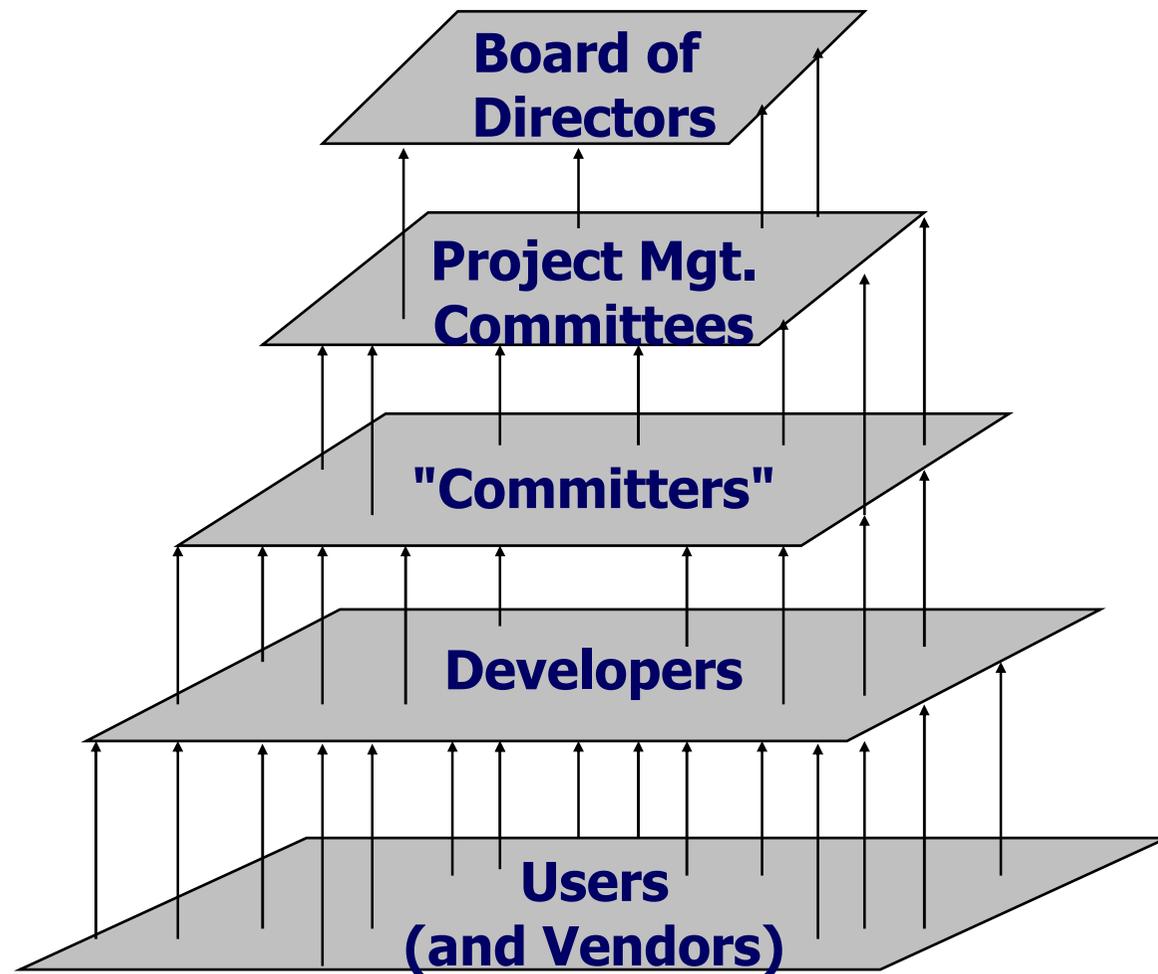
## **LINUX KERNEL**

### **Gerarchia di quattro livelli**

- ❑ Ordinary developers:** (possono proporre modifiche)
- ❑ Maintainers:** ciascuno gestisce un componente del kernel, raccolgono le modifiche e le inviano al livello superiore (t.l.)
- ❑ Trusted lieutenant:** inviano il pacchetto di modifiche al livello superiore (b.d.)
- ❑ Benevolent dictator:** è l'integratore ed arbitro delle modifiche (oltre a definire le linee guida generali)

**Test ad ogni livello**

**Compresenza di versione stabile e di sviluppo**

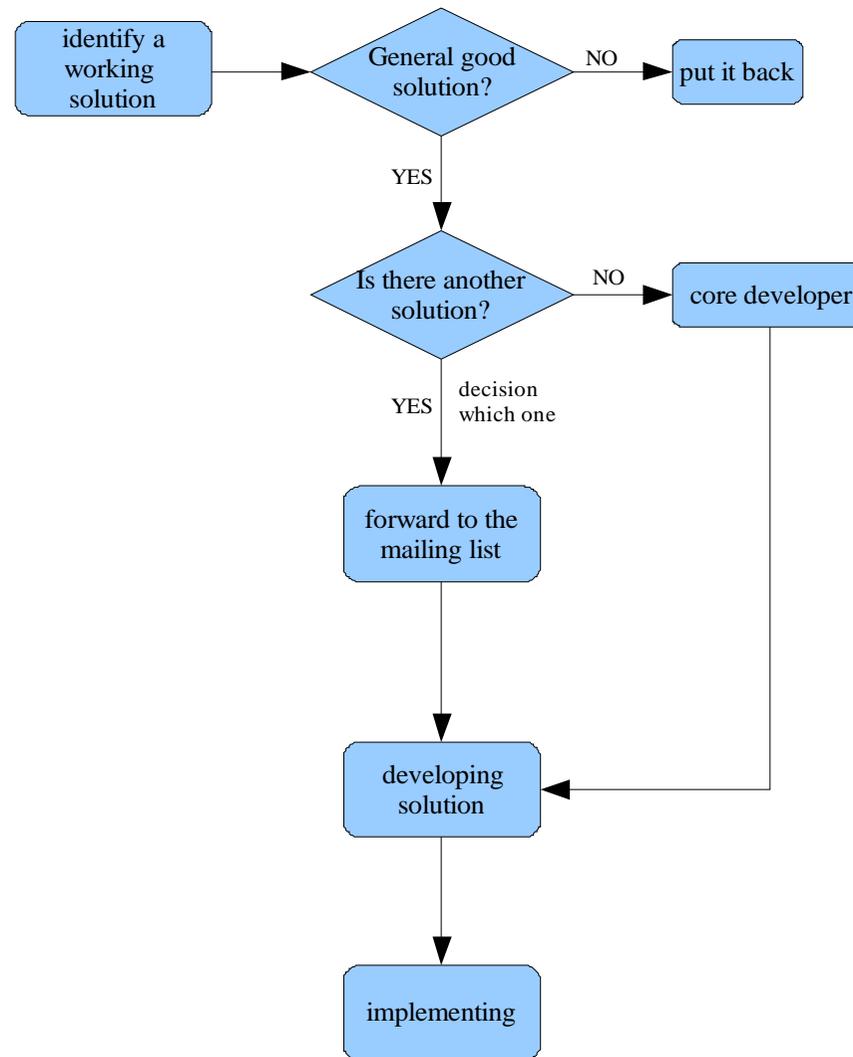


## APACHE

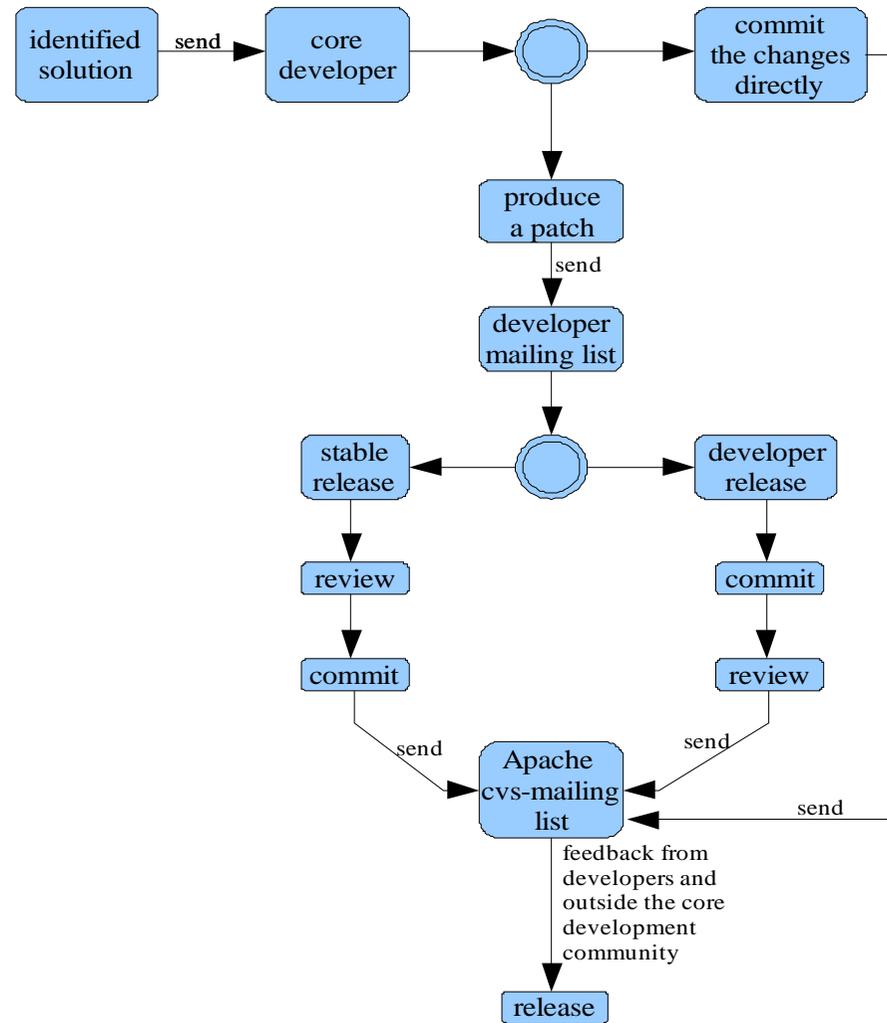
### GESTIONE DI GRUPPO

- PMC (Project Management Committee) responsabile della gestione**
  - **accesso ad invito e approvazione di tutti**
  - **espulsione per voto unanime**
  - **modifiche approvate per consenso o maggioranza**
- nuove idee: review-then-commit**
- patches: commit-then-review**
- modifiche non certe, di larga portata e nuove funzionalità devono essere prima discusse**

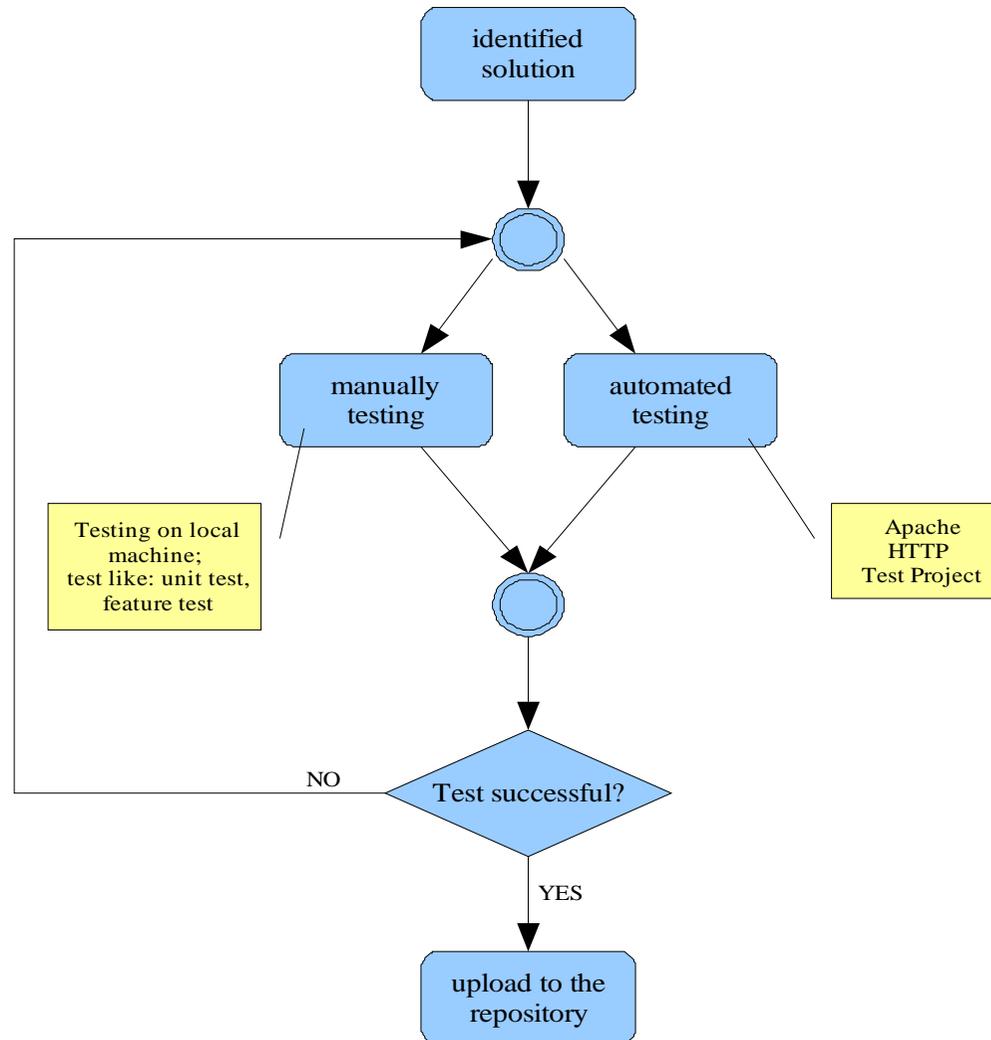
## APACHE WORKFLOW



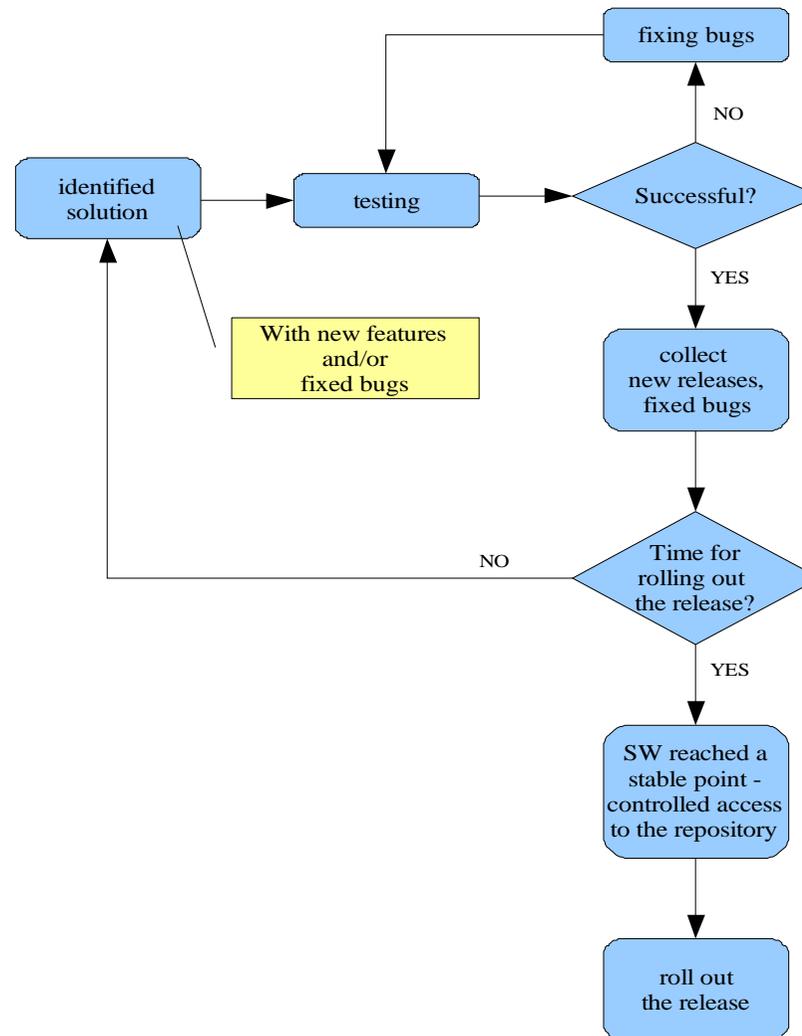
## APACHE RELEASE DEVELOPMENT



## APACHE TESTING



## APACHE QUALITY ASSURANCE



## PERL

- ❑ **Chief integrator (patch pumpkin) autorizzato a consolidare le modifiche**
- ❑ **Larry Wall (sviluppatore originale) mantiene il diritto di veto**

## SOURCEFORGE STYLE

### Configuration managemt con CVS

- ❑ Chi ha autorizzazioni di check-in effettua le modifiche
- ❑ Gli altri inviano le modifiche nelle mailing list o utilizzano il bug-tracker e ne richiedono il consolidamento

**Pochi committers → Scarsi conflitti**

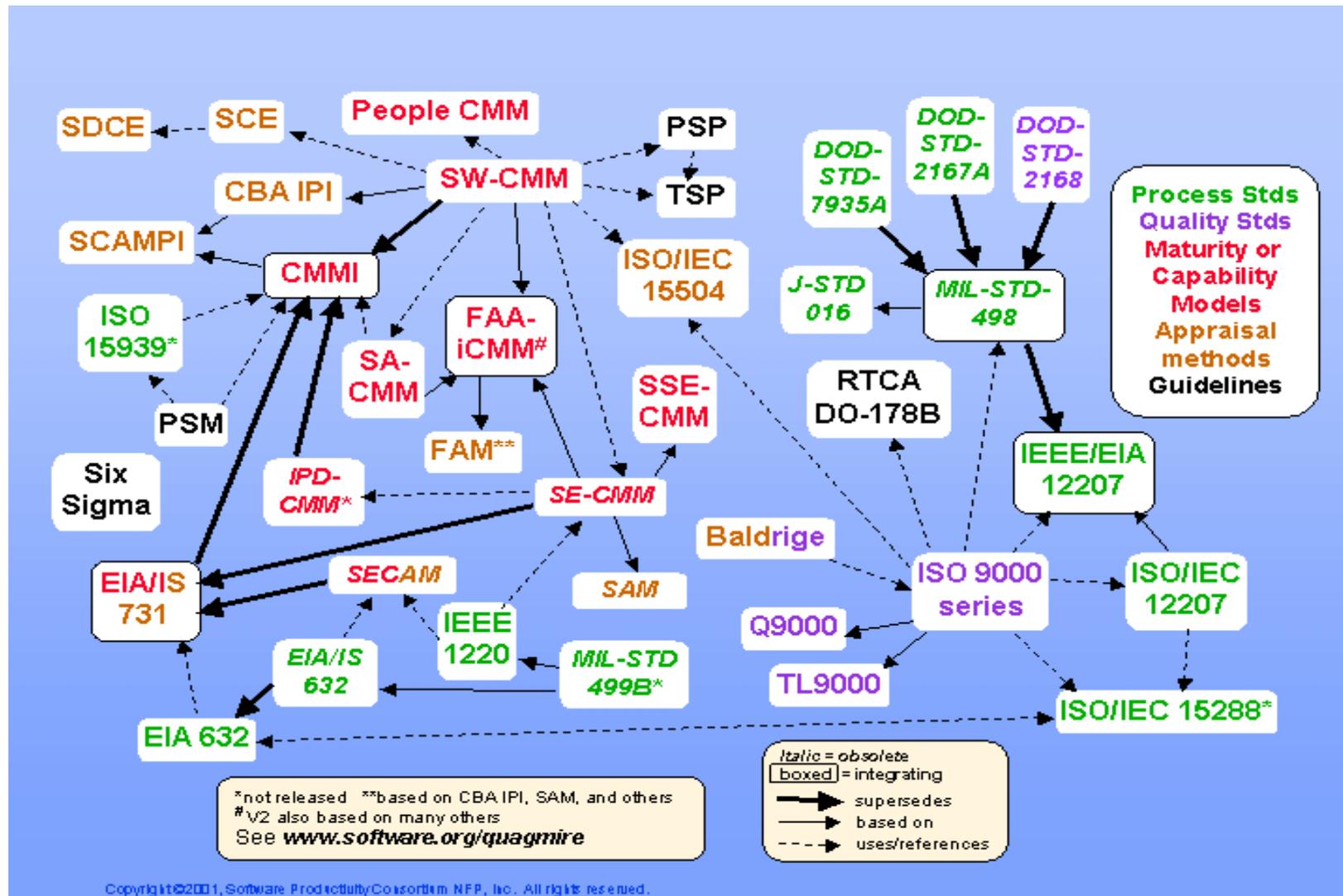
**La Qualità di un prodotto o di un servizio software può essere definita come la capacità di soddisfare requisiti espliciti e non**

**Norma ISO 8402**

- Diversi fattori di qualità**  
Correttezza, affidabilità, robustezza, portabilità, ...
- Qualità interne**  
Intrinseche alla struttura del software
- Qualità esterne**  
Percepite dagli utenti

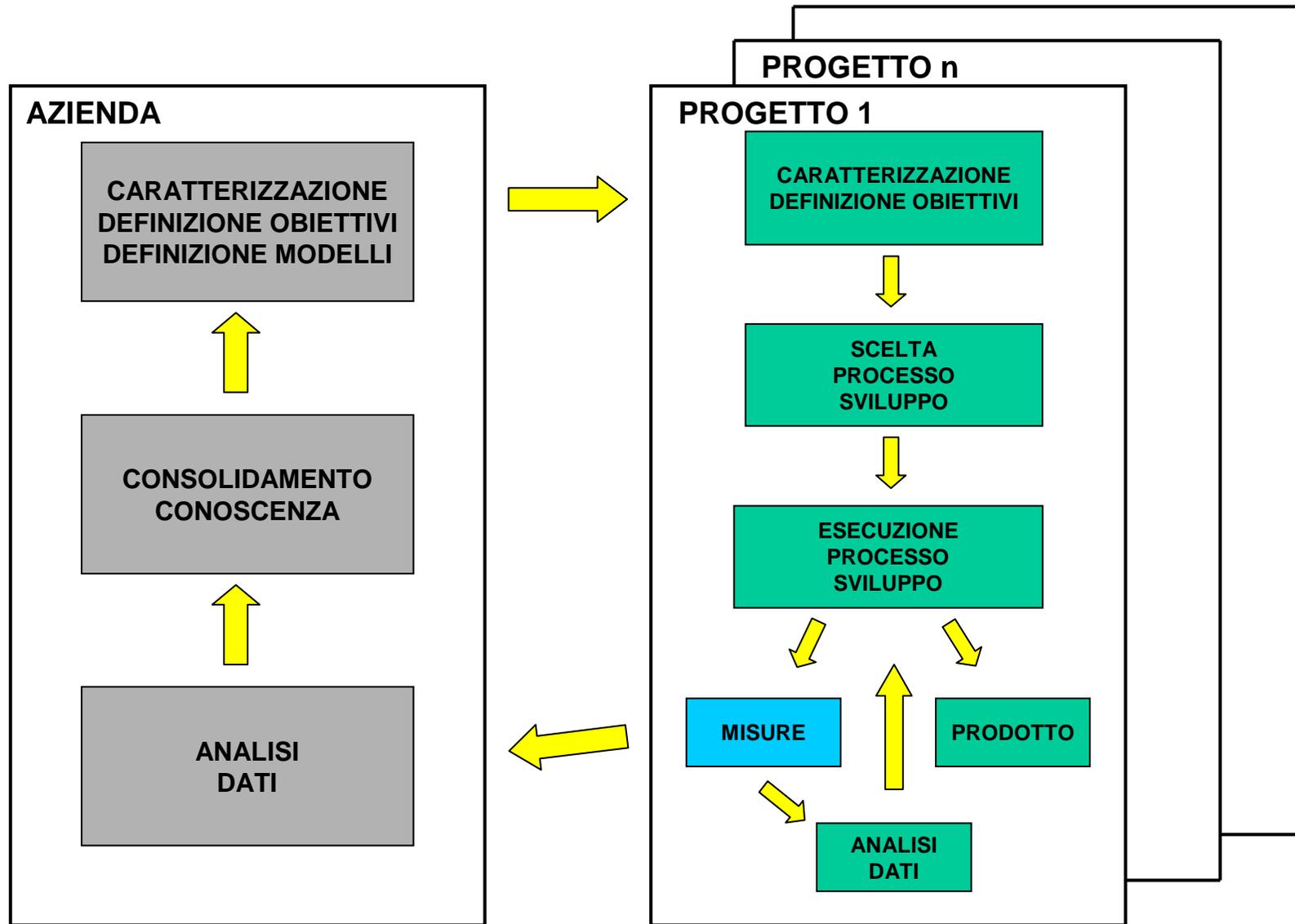
**La qualità si riferisce  
sia al prodotto sia al processo**

- ❑ **Sviluppare prodotti di alta qualità**
  - Possibile solo se il processo è di elevata qualità
  - Non si può fornire qualità a posteriori
  
- ❑ **Garantire gli obiettivi di business**
  - Non esiste una definizione univoca di qualità
  - La qualità è strettamente legata alla *mission* dell'azienda
  
- ❑ **Progettare un processo che assicuri la qualità desiderata al prodotto (e promessa al mercato)**
  - Non esiste un processo universalmente adottabile
  - Il processo deve essere coerente con gli obiettivi di business
  - Il processo deve garantire la “qualità promessa”
  - Il processo deve garantire il suo stesso “miglioramento”



The Frameworks Quagmire (fonte: [www.software.org/quagmire](http://www.software.org/quagmire))

# Ciclo del miglioramento



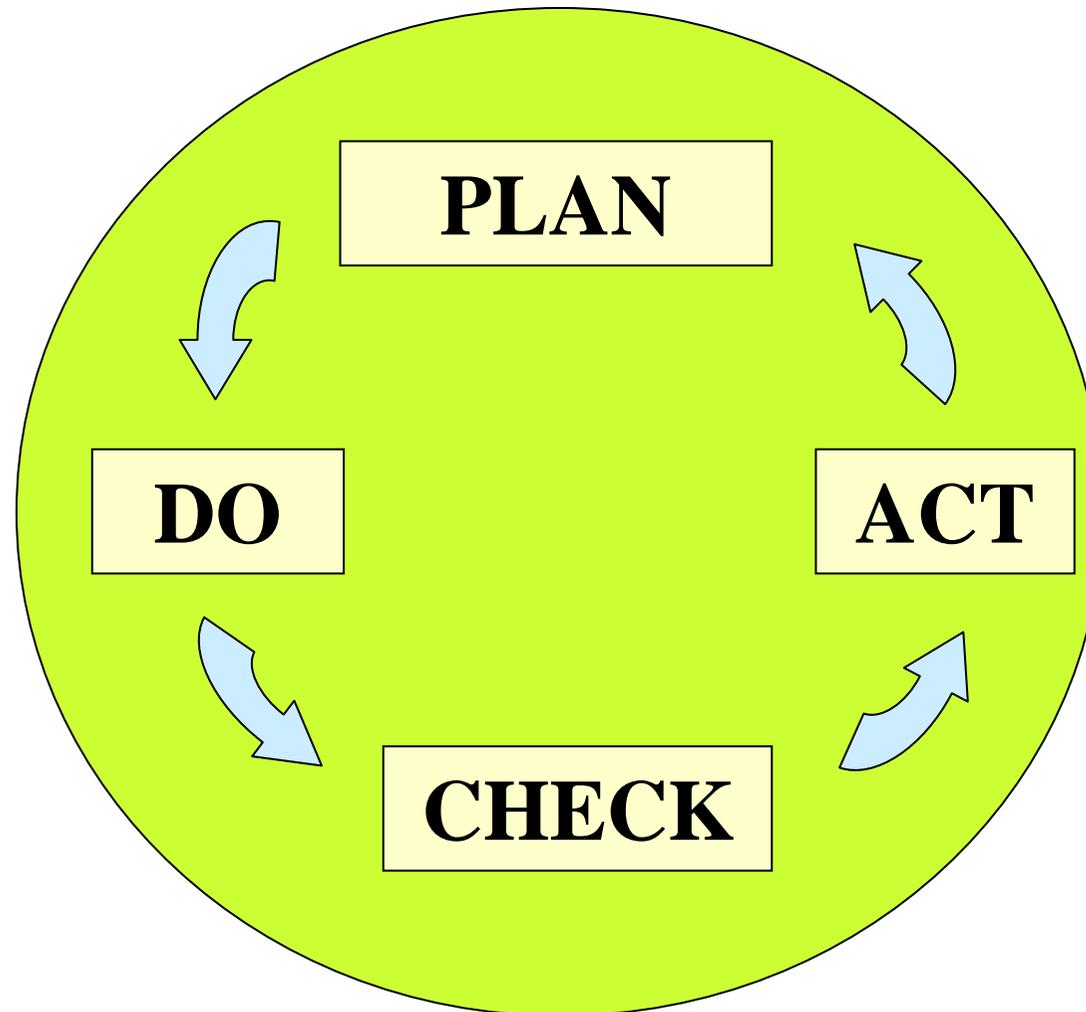


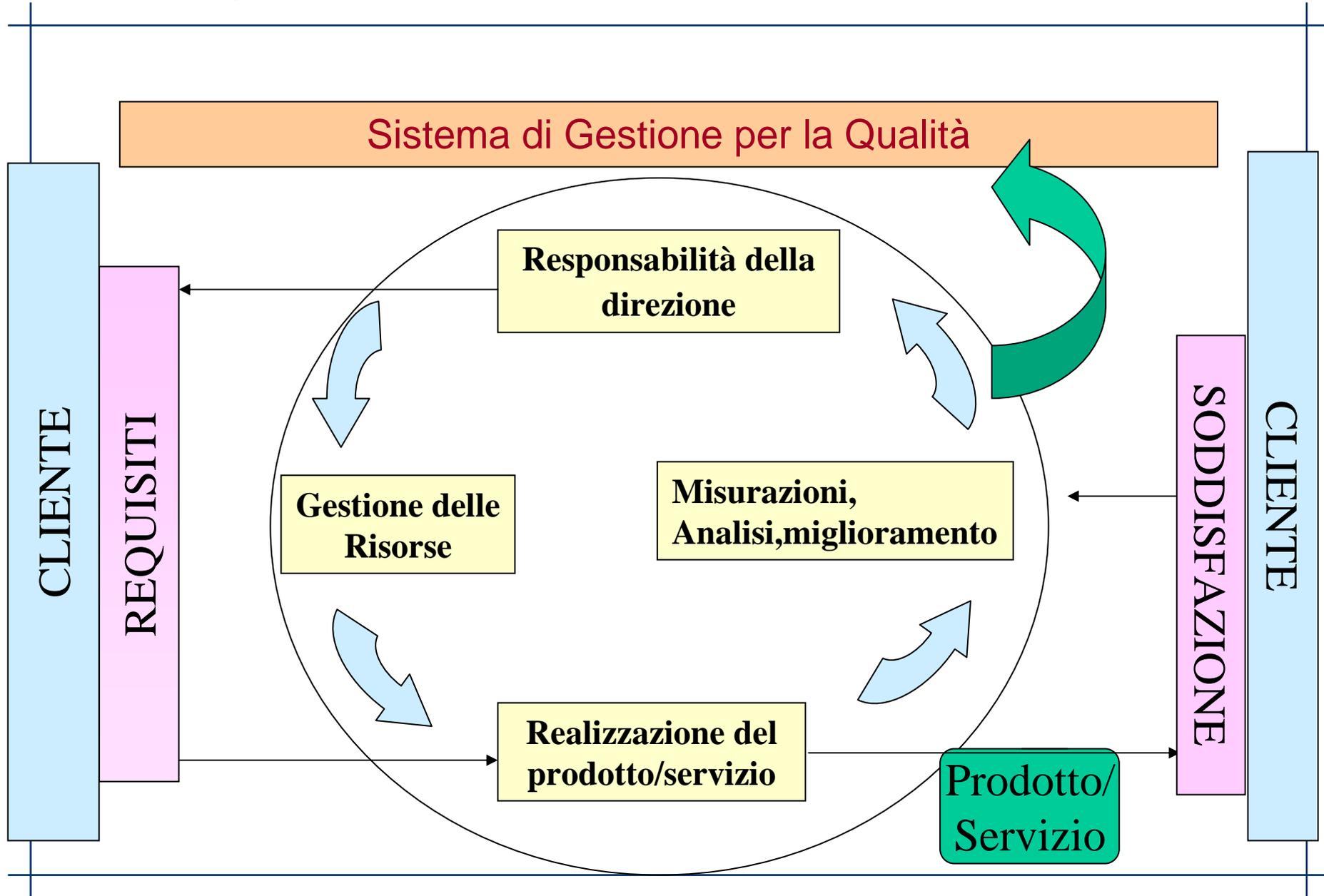
## Commonly Implemented ISO 9000 Standards

Standards and Guidelines	Purpose
ISO 9000:2000, <i>Quality management systems - Fundamentals and vocabulary</i>	Establishes a starting point for understanding the standards and defines the fundamental terms and definitions used in the ISO 9000 family which you need to avoid misunderstandings in their use.
ISO 9001:2000, <i>Quality management systems - Requirements</i>	This is the requirement standard you use to assess your ability to meet customer and applicable regulatory requirements and thereby address customer satisfaction.  It is now the only standard in the ISO 9000 family against which third-party certification can be carried.
ISO 9004:2000, <i>Quality management systems - Guidelines for performance improvements</i>	This guideline standard provides guidance for continual improvement of your quality management system to benefit all parties through sustained customer satisfaction.

## Otto principi:

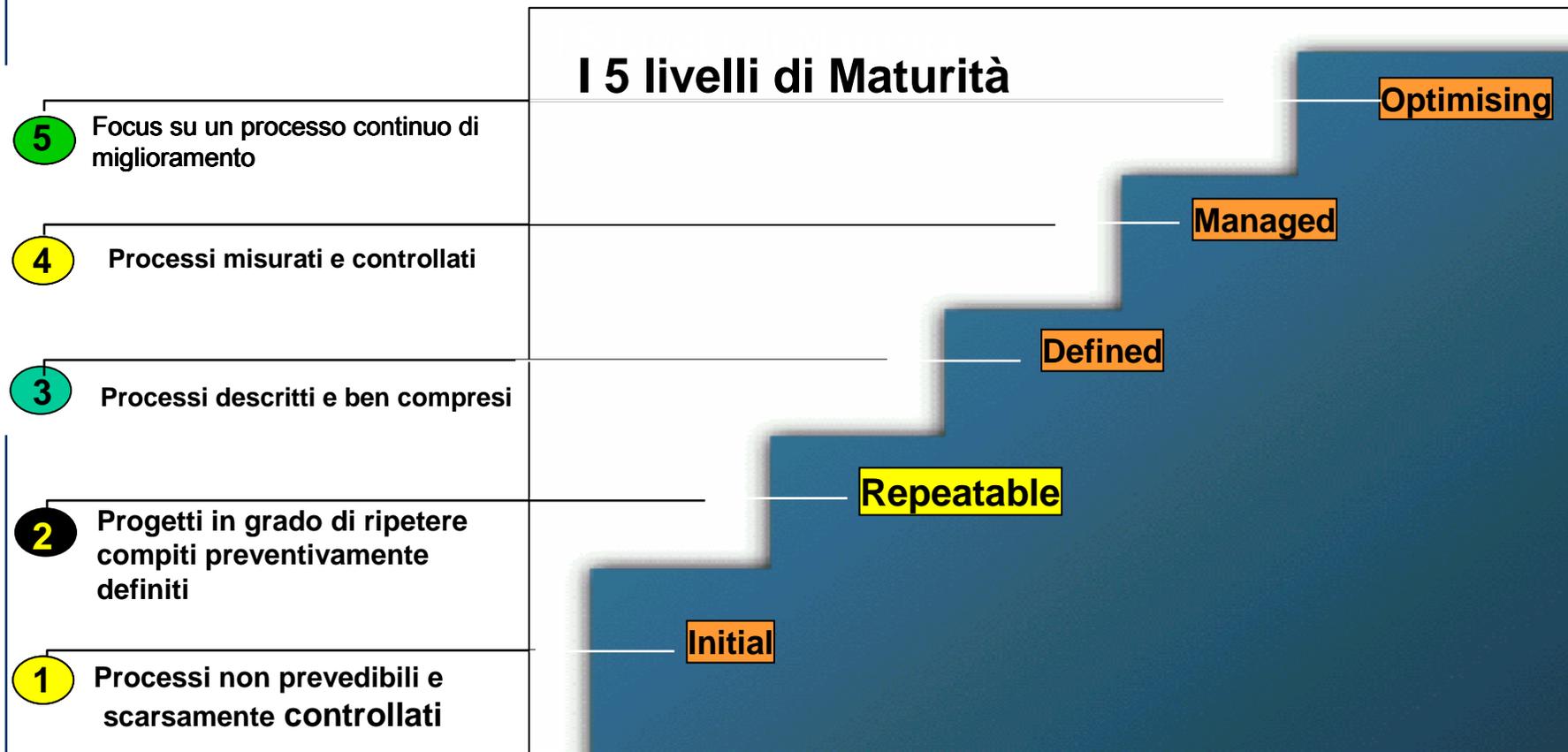
- orientamento al Cliente
  - leadership
  - coinvolgimento del personale
  - approccio per processi
  - approccio sistemico alla gestione
  - miglioramento continuo
  - processo decisionale
  - interdipendenza con i fornitori
- Coinvolgimento di tutta la struttura aziendale**

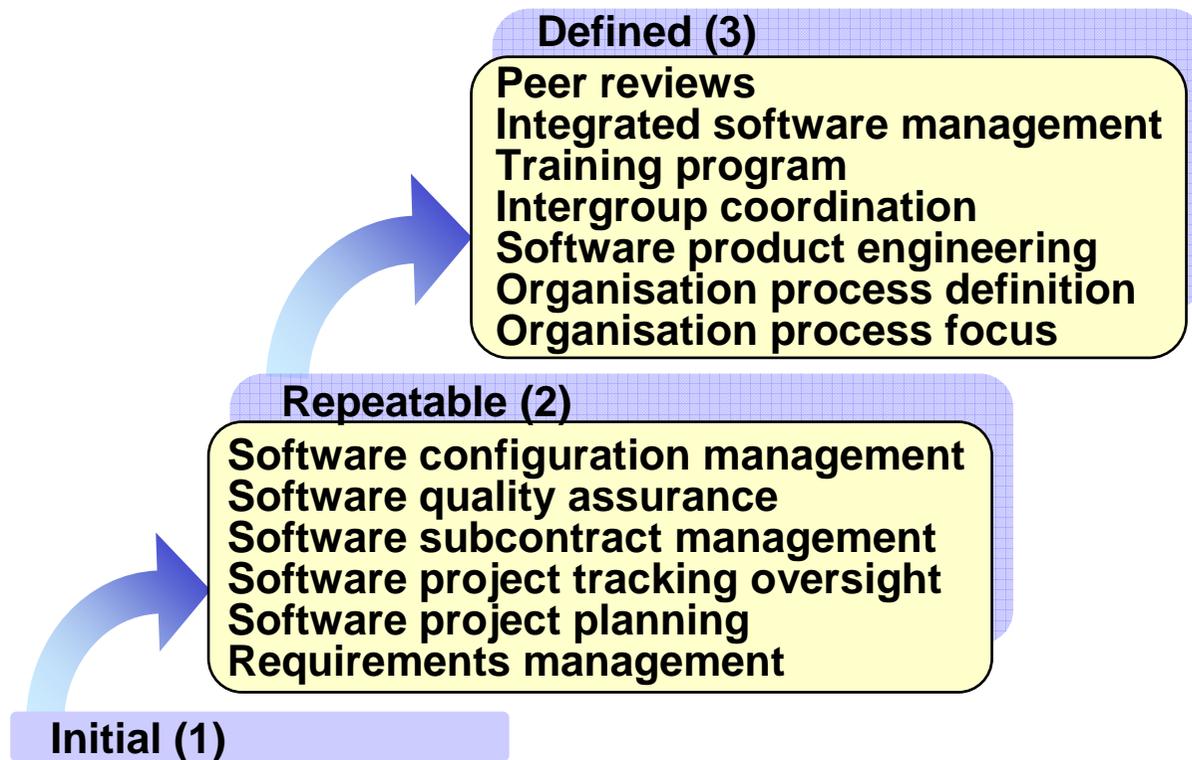




- ❑ Il **CMM** (Capability Maturity Model) definisce un modello dei processi di sviluppo del software e un insieme di regole per il loro miglioramento.
- ❑ E' stato sviluppato nel 1993 dal SEI, Software Engineering Institute per conto del DOD (Department Of Defense, USA).
- ❑ Il modello CMM permette di valutare le organizzazioni che operano nel settore IT in funzione del loro livello di “maturità” e della “capability” dei processi che attuano.

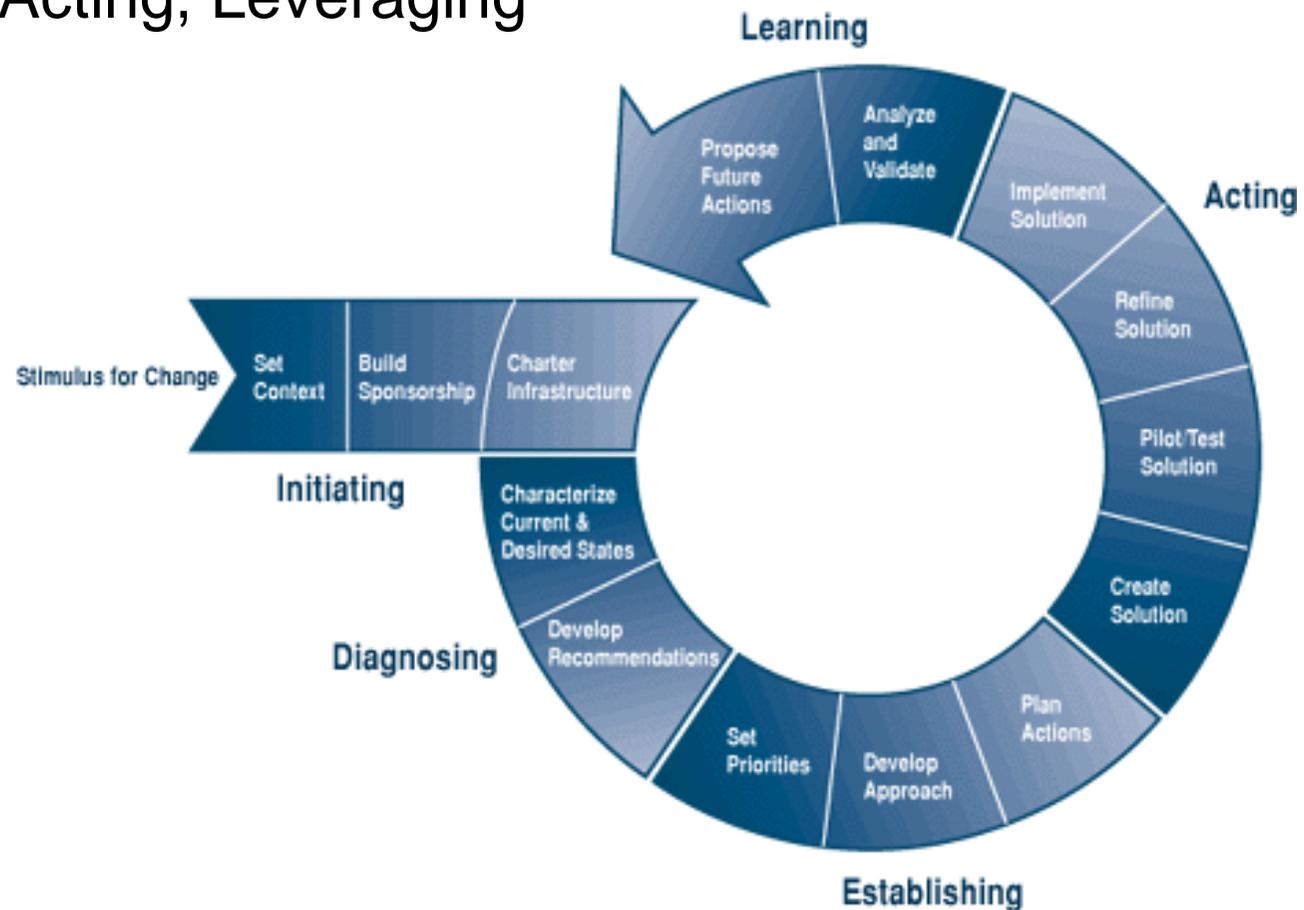
**Capability** (qualità attesa dei risultati di un processo) **Maturity** (capacità di garantire la qualità dei prodotti in uscita dai processi)  
**Model**





## IDEAL Model

Iniziating, Diagnosing, Establishing  
Acting, Leveraging



**CMM è una struttura che descrive gli elementi chiave di un efficace processo di produzione del software e quindi un percorso evolutivo di miglioramento *per le organizzazioni che realizzano software* per passare da un processo immaturo e ad hoc ad uno maturo e disciplinato**

- ❑ ISO 9001 definisce uno stadio cui tendere, senza indicare il percorso**
- ❑ CMM definisce anche i passi del percorso di miglioramento.**

***Una organizzazione ISO 9001 compliant rispetta i requisiti del livello 1 CMM e parte di quelli dei livelli 2 e 3.***



Un Framework di miglioramento continuo

Six Sigma è una metodologia rigorosa e quantitativa che mette a disposizione un insieme di tools e di practices di project management practices per l'identificazione e la riduzione della variabilità di progetto.

Le attività che devono essere seguite per migliorare le aree di underperforming e per risolvere “problemi” di qualità, sono:

- Misura della frequenza degli errori nei processi in esercizio
- Analisi dei livelli di performance correnti
- Sviluppo di procedure per migliorare la qualità dei processi esistenti
- Monitoraggio dei livelli di performance della qualità



## Un framework di controllo per l'IT

Control Objectives for Information and related Technology (COBIT) fornisce un set di best practices e strumenti per l'auditing dei processi IT rispetto alla conformità a standard, maturità e rischi.

- Disponibilità (Livello di Servizio garantito)**
- Controllo**
- Efficacia sui costi**
  
- Definito in UK negli anni 90
- Un set di specifiche a supporto degli IT managers per realizzare servizi di “buon livello” da erogare agli utenti
  - Best practices per la gestione di servizi IT che indirizzano un approccio di qualità per ottenere efficacia ed efficienza nell'utilizzo dei sistemi informativi.
  - ITIL si basa sull'esperienza collettiva acquisita a livello internazionale in attività commerciali e governative, che è stata “distillata” in un parroccio consistente e coerente oggi largamente utilizzato con uno standard de-facto.

*<http://www.itil.com>*

## **Service Support**

- Service Desk
- Incident Management
- Problem Management
- Configuration Management
- Change Management
- Release Management

## **Service Delivery**

- Availability Management
- Service Level Management
- Capacity Management
- IT Service Continuity Management
- Financial Management for IT services

## **Security**

## ❑ Service Desk

- The Service Desk provides a vital day-to-day contact point between Customers, Users, IT services and third-party support organisations. Service Level Management is a prime business enabler for this function. At an operational level, its objective is to provide a single point of contact to provide advice, guidance and the rapid restoration of normal services to its Customers and Users

## ❑ Incident Management

- The primary goal of the **incident** Management process is to restore normal service operation as quickly as possible and minimise the adverse **impact** on business operations, thus ensuring that the best possible levels of service quality and **availability** are maintained. 'Normal service operation' is defined here as service operation within **Service Level Agreement (SLA)** limits.

## ❑ Problem Management

- The goal of **Problem** Management is to minimise the adverse **impact** of **incidents** and Problems on the business that are caused by errors within the IT Infrastructure, and to prevent recurrence of Incidents related to these errors. In order to achieve this goal, Problem Management seeks to get to the root cause of Incidents and then initiate actions to improve or correct the situation. The **Problem** Management process has both reactive and proactive aspects. The reactive aspect is concerned with solving Problems in response to one or more **incidents**. Proactive Problem Management is concerned with identifying and solving Problems and **known errors** before Incidents occur in the first place

## ❑ Configuration Management

- The goals of Configuration Management are to:
  - account for all the IT assets and configurations within the organisation and its services.
  - provide accurate information on configurations and their documentation to support all the other Service Management processes.
  - provide a sound basis for **Incident** Management, **Problem** Management, **Change** Management and **Release** Management.
  - verify the configuration records against the infrastructure and correct any exceptions.

## ❑ Change Management

- The goal of the Change Management Process is to ensure that standardised methods and procedures are used for efficient and prompt handling of all Changes, in order to minimise the **impact** of Change-related **incidents** upon service quality, and consequently to improve the day-to-day operations of the organisation.

## ❑ Release Management

- The goals of Release Management are:
  - to plan and oversee the successful rollout of software and related hardware
  - to design and implement efficient procedures for the distribution and installation of Changes to IT systems
  - to ensure that hardware and software being changed is traceable, secure and that only correct, authorised and tested versions are installed
  - to communicate and manage expectations of the Customer during the planning and rollout of new Releases
  - to agree the exact content and rollout plan for the Release, through liaison with **Change Management** - to implement new software Releases or hardware into the operational environment using the controlling processes of **Configuration Management** and Change Management – a Release should be under Change Management and may consist of any combination of hardware, software, firmware and document C
  - to ensure that master copies of all software are secured in the Definitive software library (DSL) and that the Configuration management database (CMDB) is updated
  - to ensure that all hardware being rolled out or changed is secure and traceable, using the services of Configuration Management.

## □ Availability Management

- The goal of the Availability Management process is to optimise the capability of the IT Infrastructure, services and supporting organisation to deliver a cost effective and sustained level of **Availability** that enables the business to satisfy its business objectives.

Availability Management should ensure the required level of Availability is provided. The measurement and monitoring of IT Availability is a key activity to ensure Availability levels are being met consistently. Availability Management should look continuously to optimise the Availability of the IT Infrastructure, services and supporting organisation, in order to provide cost effective Availability improvements that can deliver evidenced business and User benefits.

## □ Security

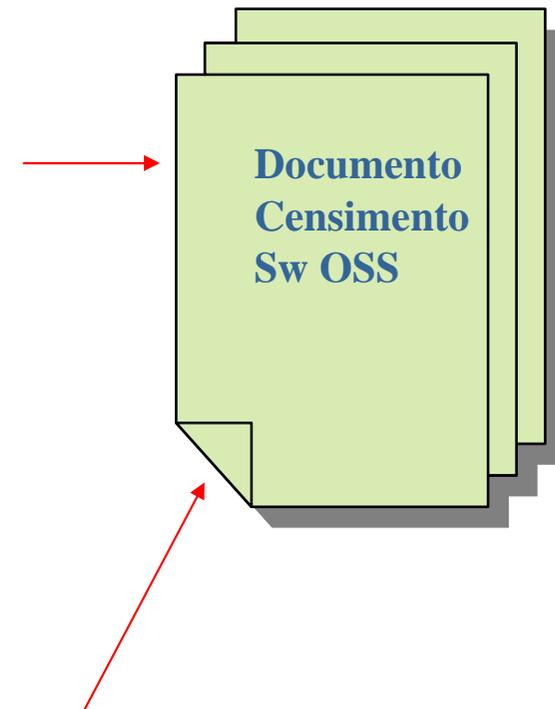
- Security is a global subject, applying to every aspects of Network Operations.
- The goal of Security Management is to ensure by specific configurations or actions, the usage and the operation of the Network will be limited to people (group of people / communities) explicitly known and measures will prevent other people to have access.

	Process Oriented Control and Audit System	Process Improvement Methodology	IT Best Practices
ITIL			
SixSigma			
COBIT			

 Not Addressed by Framework  
 Strength of Framework

## Osservatorio FOSS Engineering

- Riferimenti: URL, Comunità di riferimento, Eventuale Vendor
- Licenza (tipo ed eventuale presenza di soluzione commerciale)
- Data inizio progetto
- Ultima data aggiornamento
- Versione attuale
- Frequenza rilasci
- Livello di interesse (esempio partendo dall'utilizzo dei forum)
- Livello di maturità
- Livello di diffusione
- Livello documentazione
- Livello di usabilità (tool, skill, formazione, ..)
- Gestione retro compatibilità
- Standard riferimento
- Relazione con altri progetti OSS
- Relazione con altri progetti commerciali
- Conoscenza aziendale
- Relazioni con la comunità



- QSOS** (Atos Origin)
- Open Source Maturity Model** (Navica)
- Open Source Maturity Model** (Cap Gemini)
- Open Source Catalogue** (Optaros)
- OSS Watch**
- IRCA** (Weehler)
- OpenBRR** (Carnegie Mellon e altri)



[www.qsos.org](http://www.qsos.org)



**Realizzato da Atos Origin, rilasciato in licenza GNU FDL**

[www.navicasoft.com/pages/osmm.htm](http://www.navicasoft.com/pages/osmm.htm)

## **Tre fasi:**

- Assess Maturity Element**
- Assign Weighting Factor**
- Calculate Product Maturity Score**

## **Parametri:**

- Prodotto**
- Supporto**
- Documentazione**
- Formazione**
- Integrazione**
- Servizi Professionali**

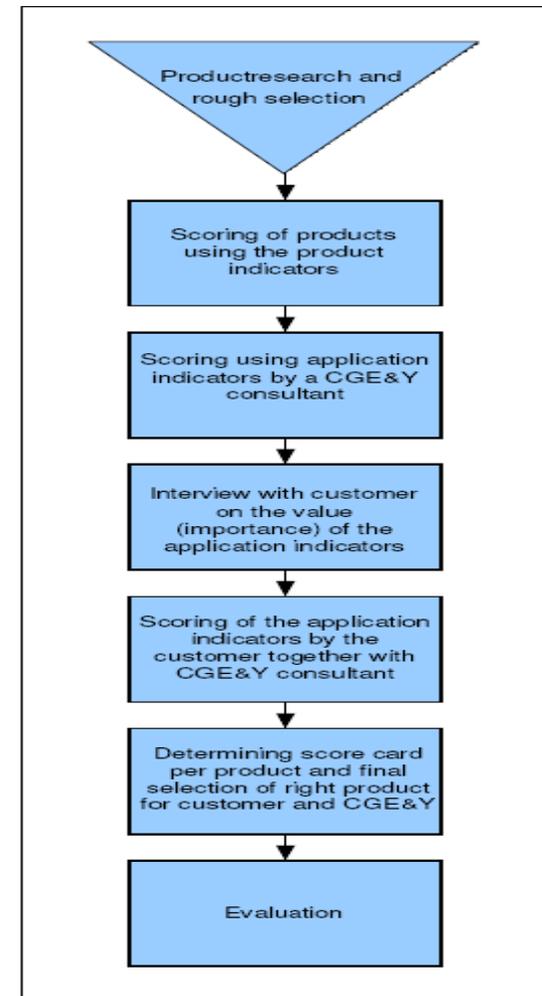
**punteggio tra 0 e 100**

[www.seriouslyopen.org/nuke/html/modules.php?name=News&file=categories&op=newindex&catid=2](http://www.seriouslyopen.org/nuke/html/modules.php?name=News&file=categories&op=newindex&catid=2)

**Product indicators**

Indicator	Product A	Product B
<b>Product</b>		
Age	1	3
Licensing	2	2
Human hierarchies	1	2
Selling points	2	1
Developer community	2	2
<b>Integration</b>		
Modularity	1	1
Collaboration with other products	2	2
Standards	2	2
<b>Use</b>		
Support	2	3
Ease of deployment	3	4
<b>Acceptance</b>		
User community	2	3
Market penetration	3	2
<b>TOTAL</b>	<b>23</b>	<b>27</b>

**7 fasi, 27 parametri**



[www.optaros.com](http://www.optaros.com)

## Quattro criteri:

- Funzionalità** (excellent, fair, poor)
- Comunità** (excellent, ok, poor)
- Maturità** (very mature, ok, poor)
- Trend** (crescita, stabile, declino)

## Enterprise Readiness Rating (maturo, ok, non raccomandato)

### Ulteriori informazioni:

- Versione**
- Descrizione** (url)
- Licenza**
- Supporto**

Realizzato da Optaros

[www.oss-watch.ac.uk/resources/tips.xml](http://www.oss-watch.ac.uk/resources/tips.xml)

## Undici elementi:

- Reputazione
- Effort di sviluppo
- Standard e interoperabilità
- Supporto della comunità
- Supporto commerciale
- Versione
- Versione 1.0
- Documentazione
- Skill disponibile
- Licenza
- Copertura



[www.dwheeler.com/oss\\_fs\\_eval.html](http://www.dwheeler.com/oss_fs_eval.html)

### **Quattro fasi:**

- Identify candidates
- Read existing reviews
- Compare the leading's programs attributes to your needs
- Analyze the top candidates in more depth

### **Tredici attributi:**

- |   |  |
|---|--|
| <input type="checkbox"/> Funzionalità           | <input type="checkbox"/> Usabilità                 |
| <input type="checkbox"/> Costo                  | <input type="checkbox"/> Sicurezza                 |
| <input type="checkbox"/> Market share           | <input type="checkbox"/> Flessibilità/Adattabilità |
| <input type="checkbox"/> Supporto               | <input type="checkbox"/> Interoperabilità          |
| <input type="checkbox"/> Manutenzione/Longevità | <input type="checkbox"/> Licenza                   |
| <input type="checkbox"/> Robustezza             |  |
| <input type="checkbox"/> Prestazioni            |  |
| <input type="checkbox"/> Scalabilità            |  |

by David A. Wheeler

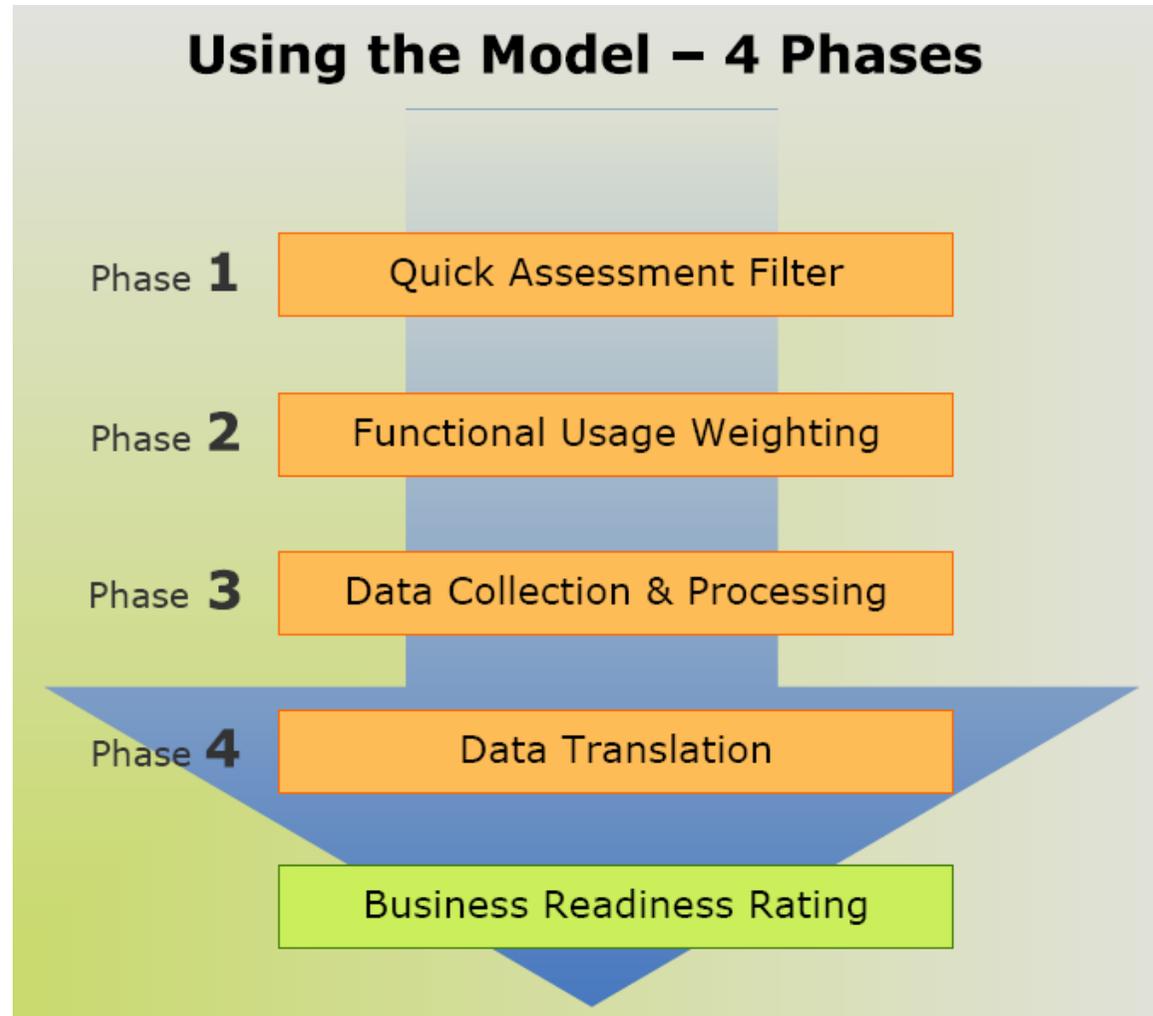
## Evaluation Categories for Open Source Software

- Functionality
- Operational software characteristics
  - Reliability, scalability, performance, usability, security, installation, interoperability
- Support and services
- Documentation
- Development process
  - Leadership, project data, task assignments\*
- Software Technology Attributes\*
  - Architecture, code quality
- Adoption
  - Size, activity level, community\*

\* Available for open source,  
but not for closed source

[www.openbrr.org](http://www.openbrr.org)





## Prima fase: Quick Assessment Filter

□ **Creazione di una lista di candidati** per un assessment più approfondito; in questa fase è consigliabile utilizzare alcuni indicatori per aiutare a selezionare i progetti più interessanti, quali:

- Tipo di licenza.
- Conformità agli standard
- Referenze
- Esistenza di una organizzazione alle spalle che supporta lo sviluppo del progetto.
- Linguaggio di programmazione utilizzato
- Supporto dell'internazionalizzazione
- Esistenza di altre valutazioni di enti esterni
- Esistenza di pubblicazioni
- Il progetto è seguito da analisti (IDC, Gartner, ..)

## Seconda fase: Functional Usage Weighting

- ❑ **Analisi approfondita delle caratteristiche dei progetti** per ordinare le categorie e assegnare la *functional orientation*.
- ❑ **Normalizzazione dei dati delle metriche** (proprietà misurabili del progetto), in particolar modo le metriche che non sono facilmente misurabili ma a cui è associata una valutazione qualitativa (che può essere soggettiva). Organizzazione delle metriche in aree di interesse indicate come categorie a cui associare un peso.
- ❑ **Identificazione dell'ambito di utilizzo del progetto** (sperimentazione, ambiente di produzione, ...) **e della tipologia** (web browser, web server, office, ...), da cui derivare la *functional orientation* intesa come combinazione del tipo e ambito e identificata come “peso associato alla singola categoria” (serve per normalizzare le metriche contenute in essa): percentuale associata ad ognuna delle 7 categorie scelte e utilizzate per la valutazione

## Terza fase: Data Collection & Processing

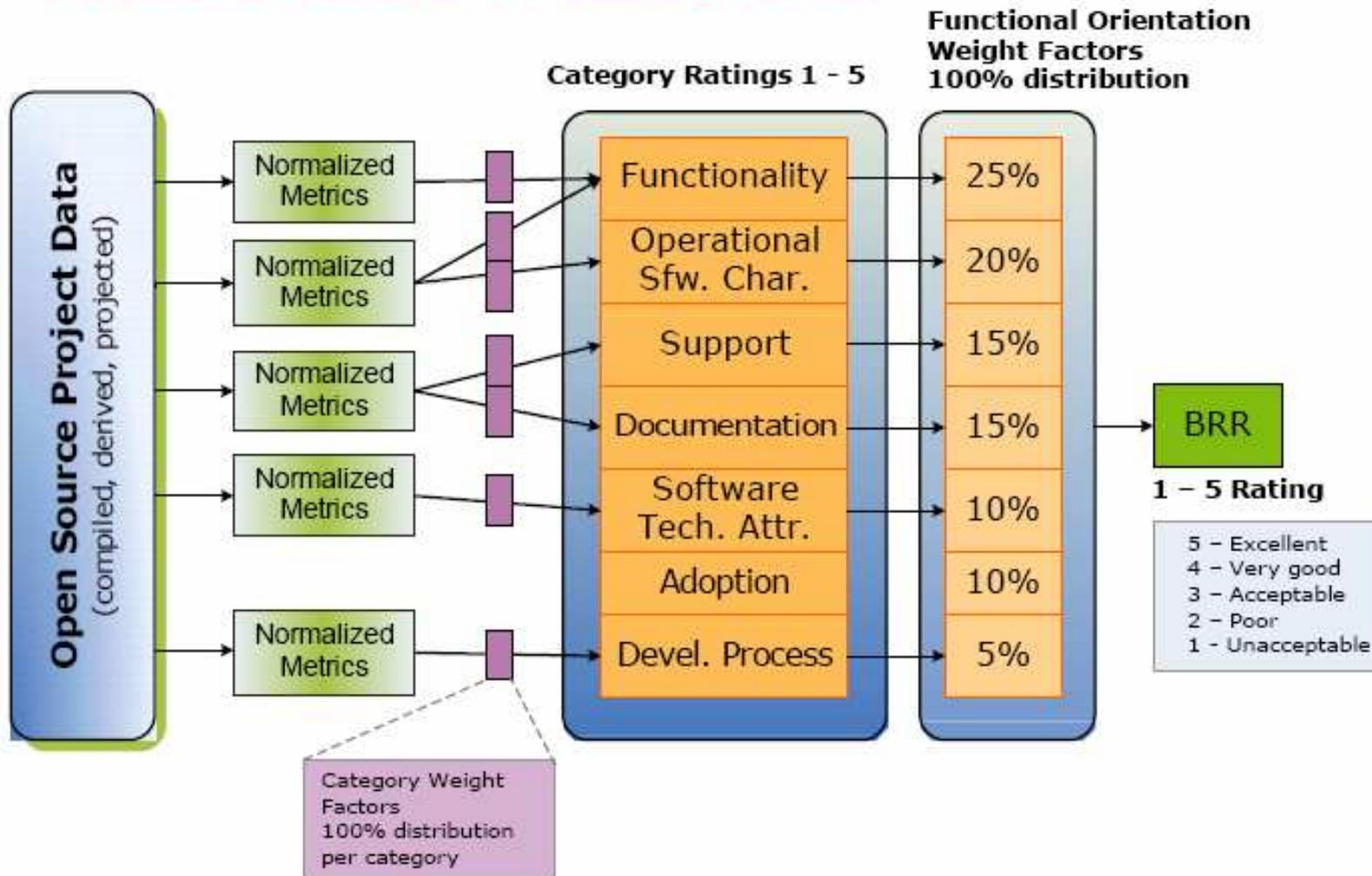
❑ **Raccolta dei dati:** per ogni categoria sono identificate un insieme di metriche che possono essere modificate (aggiungendo o togliendo metriche specifiche per la tipologia), queste sono generiche e valgono per tutti i tipi di progetti; solo la categoria “funzionale” contiene delle metriche che sono specifiche della tipologia di progetto in valutazione. I valori da assegnare alle singole metriche devono essere normalizzati tra 1 e 5; per le metriche qualitative sono proposte alcune linee guida

## Quarta fase: Data translation

### **Trasformazione dei dati nella valutazione BRR**

Terminate le fasi precedenti viene generata la valutazione finale BRR sommando le singole metriche, con risultato un numero tra 1 e 5. Per la valutazione delle metriche è disponibile un foglio MS-excel utilizzabile come template su cui aggiungere le varie metriche funzionali.

## Business Readiness Rating model



□ **Wheeler D.A., *Why Open Source Software / Free Software (OSS/FS, FLOSS or FOSS)? Look at the Numbers!*,  
[www.dwheeler.com/oss\\_fs\\_why.html](http://www.dwheeler.com/oss_fs_why.html)**

□ **Wheeler D.A., *How to Evaluate Open Source Software / Free Software (OSS/FS) Programs*,  
[www.dwheeler.com/oss\\_fs\\_\\_eval.html](http://www.dwheeler.com/oss_fs__eval.html)**