

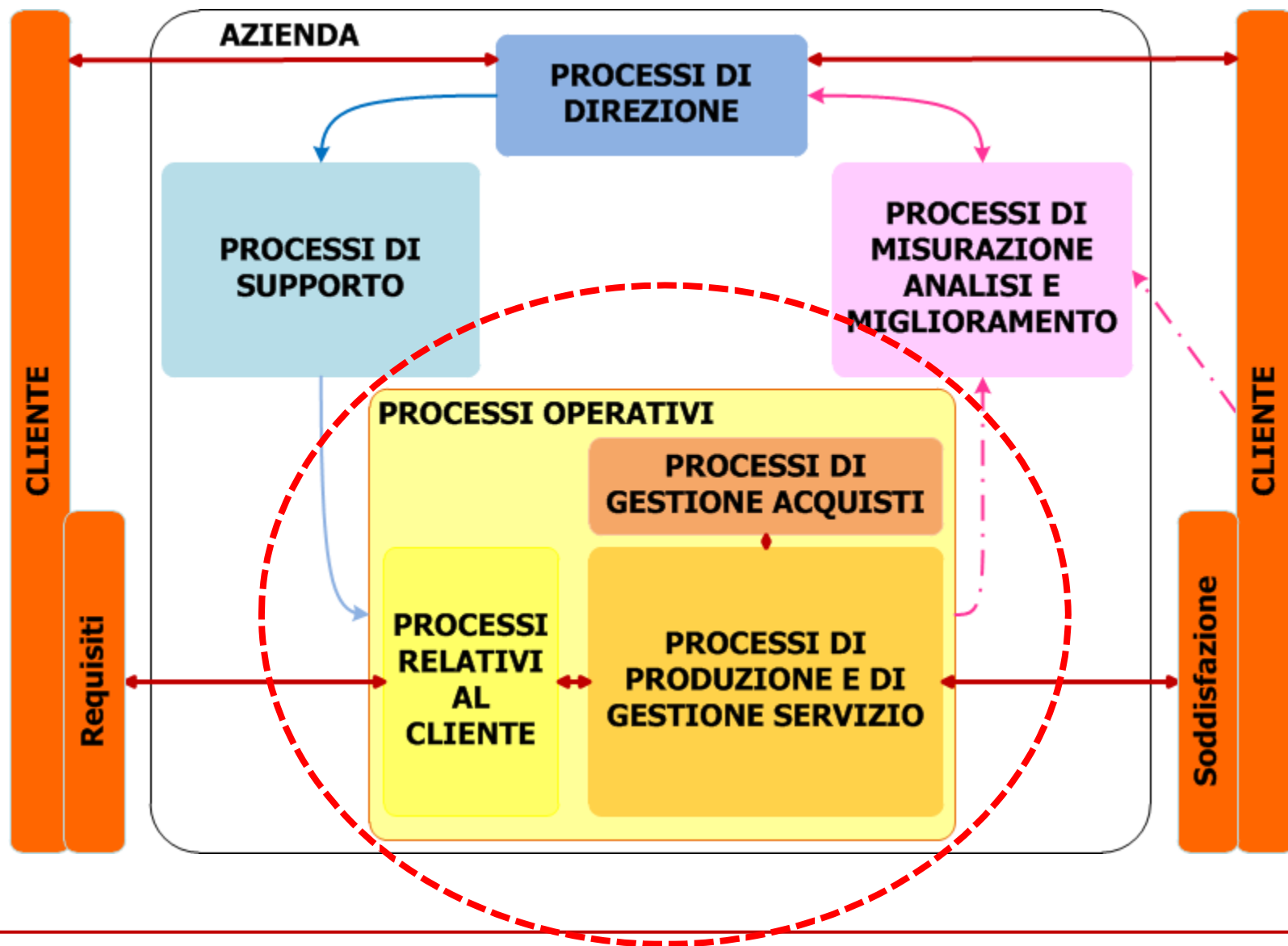
## CICLO DI VITA DEL SOFTWARE

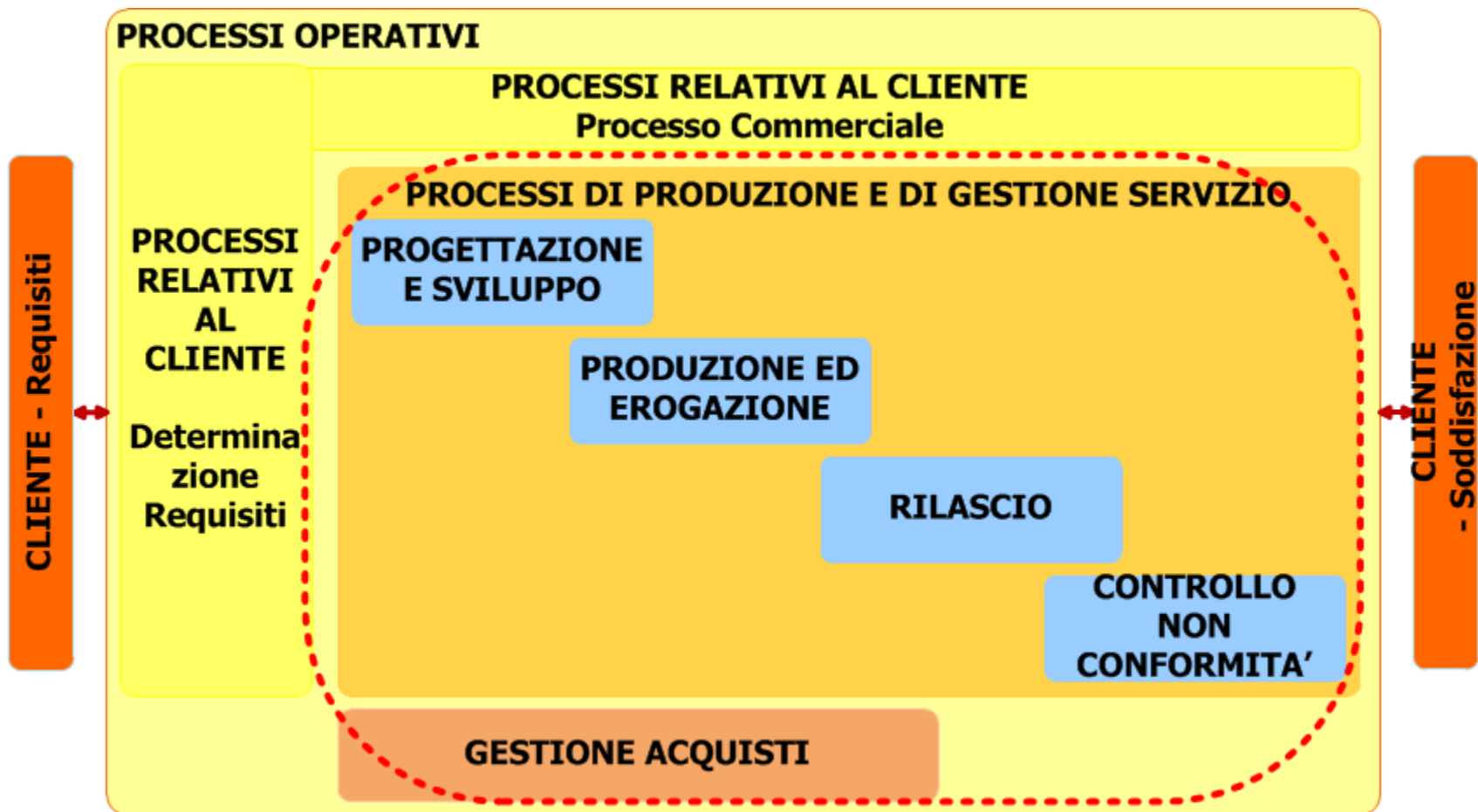
Maria Teresa Palomba – A-Key S.r.l.  
[t.palomba@a-key.it](mailto:t.palomba@a-key.it)



**Riferimenti:**

- A. <http://www.computer.org/web/swebok> - Software Engineering Body of Knowledge (SWEBOK), version 3.0
- B. <http://www.sei.cmu.edu/library/abstracts/reports/10tr033.cfm> modello CMMI-DEV v. 1.3
- C. [http://www.automotivespice.com/fileadmin/software-download/Automotive\\_SPICE\\_PAM\\_30.pdf](http://www.automotivespice.com/fileadmin/software-download/Automotive_SPICE_PAM_30.pdf)  
- Automotive SPICE PAM PRM v.3.0



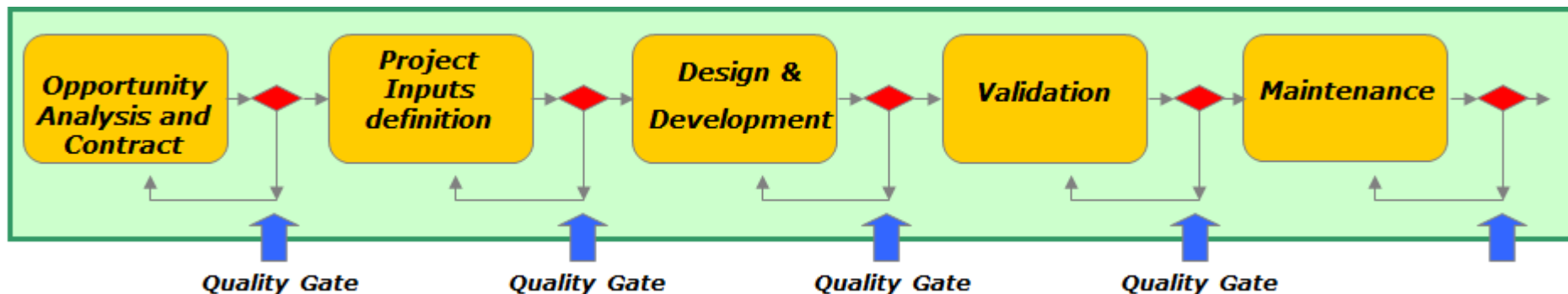


## Pianificazione della progettazione e sviluppo

L'Organizzazione deve **pianificare e tenere sotto controllo le attività di progettazione e sviluppo** dei prodotti/servizi, con riferimento a:

- ❑ fasi di svolgimento
- ❑ riesami, verifiche e validazioni
- ❑ responsabilità ed autorità

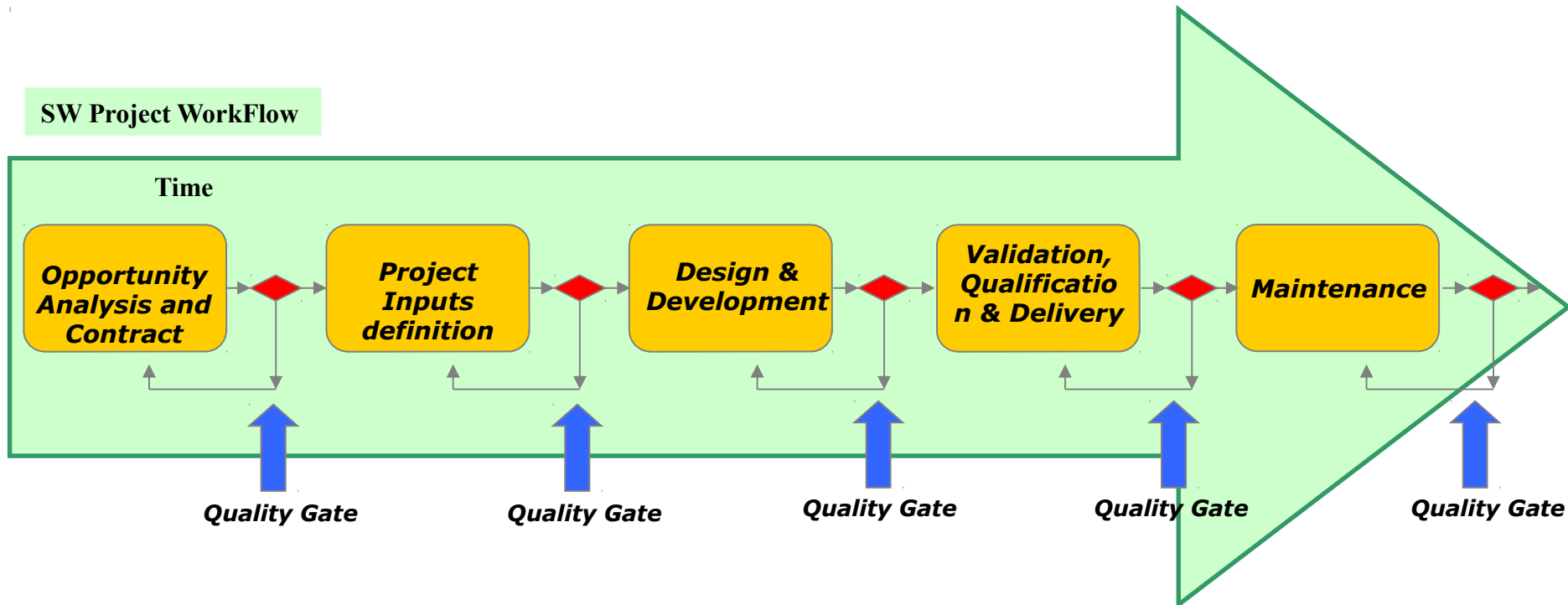
Esempio:

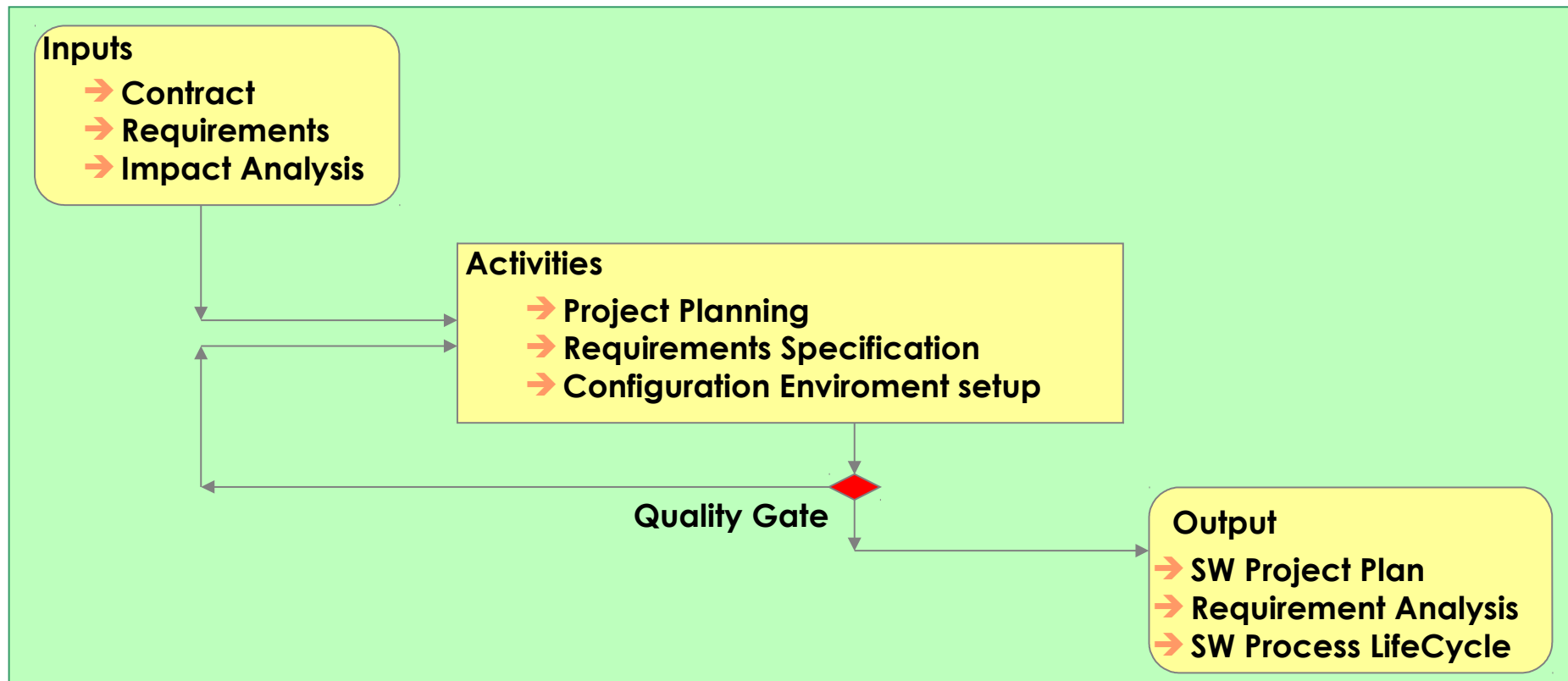
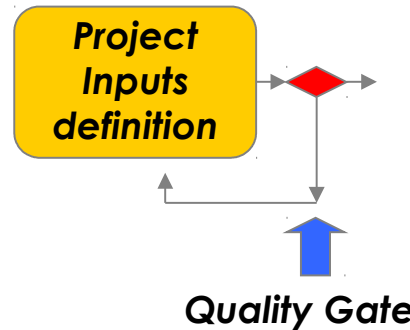


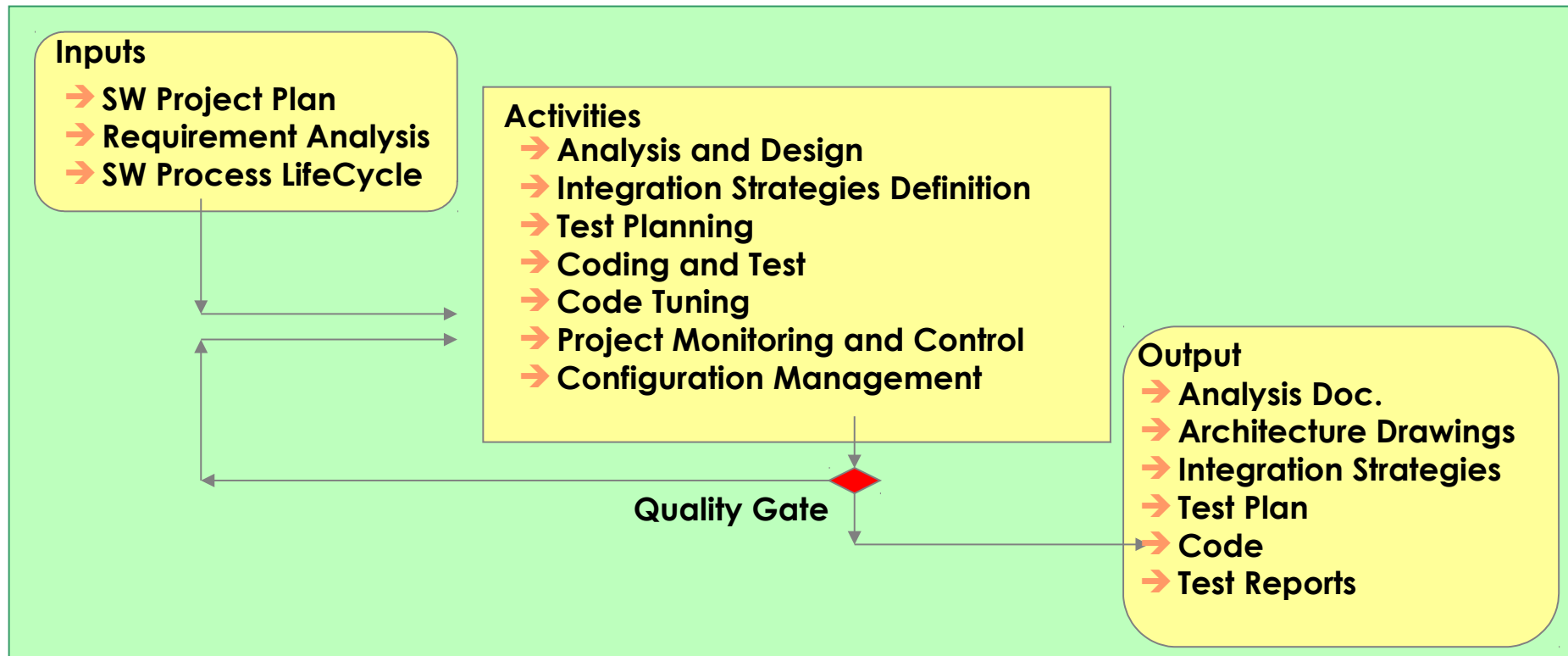
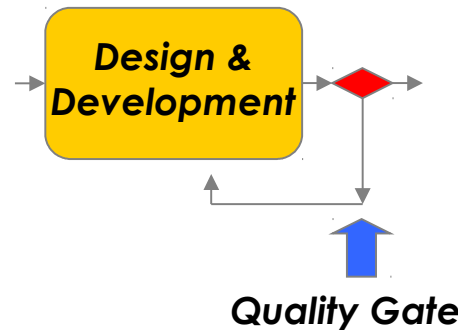
## Organizzazione in fasi principali del processo di gestione del progetto:

- Input, attività ed output definiti per ogni fase
- Output da ogni fase è Input per la fase successiva
- Attività di Verifica e controllo definite per ogni fase
- Un Riesame di fine fase (Main Quality Gate) è il check-point per il completamento della fase

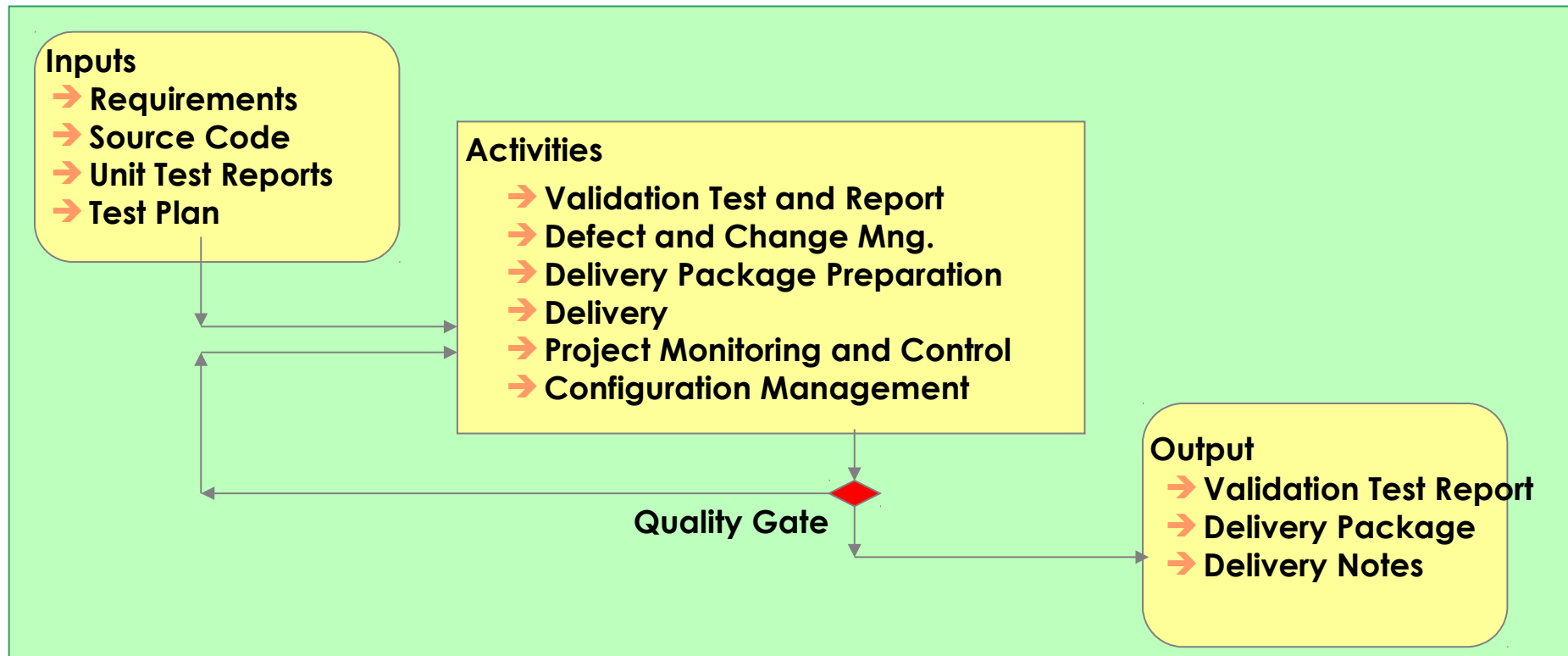
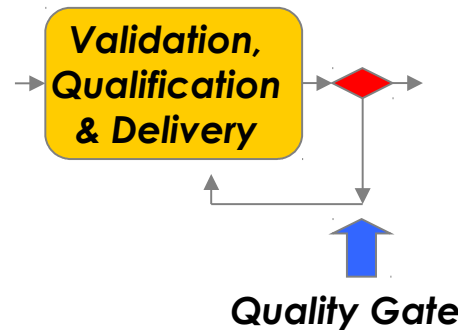
SW Project WorkFlow

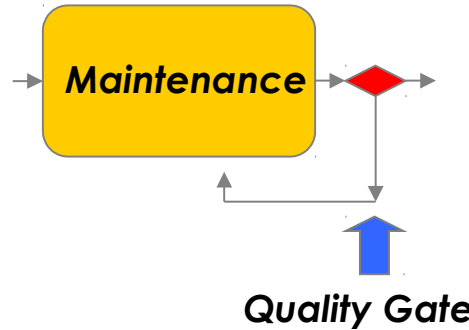




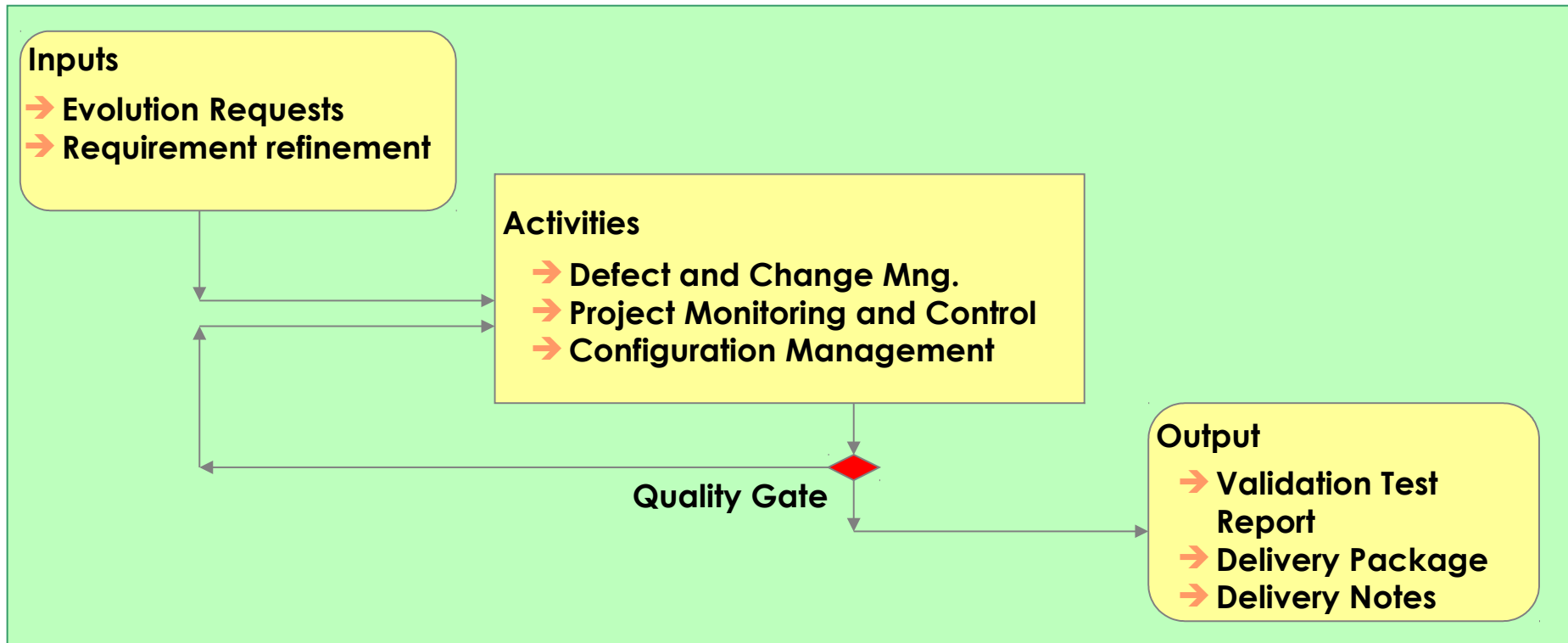








### Workflow



## Modelli di ciclo di vita del SW *Rif. A SWEBOK*

“The intangible and malleable nature of software permits a wide **variety of software development life cycle models**, ranging

- from linear models in which the phases of software development are accomplished sequentially with feedback and iteration as needed followed by integration, testing, and delivery of a single product;
- to iterative models in which software is developed in increments of increasing functionality on iterative cycles;
- to agile models that typically involve frequent demonstrations of working software to a customer or user representative who directs development of the software in short iterative cycles that produce small increments of working, deliverable software.”

## Modelli di ciclo di vita del SW *Rif. A SWEBOOK*

“Incremental, iterative, and agile models can deliver early subsets of working software into the user environment, if desired.

**Linear SDLC** models are sometimes referred to as **predictive** software development life cycle models, while **iterative and agile SDLCs** are referred to as **adaptive** software development life cycle models.

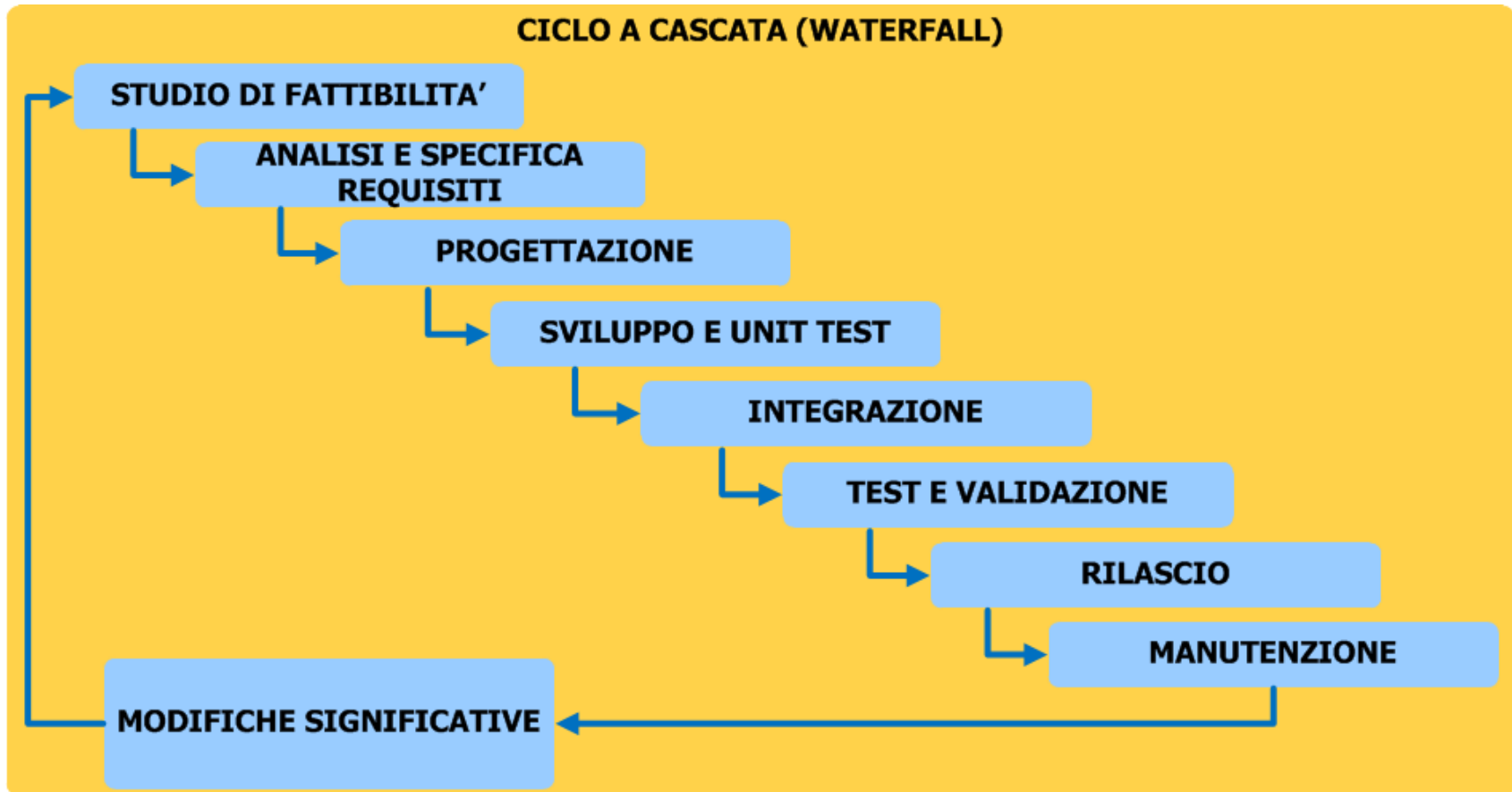
It should be noted that various **maintenance activities during an SPLC can be conducted using different SDLC models**, as appropriate to the maintenance activities.

## Modelli di ciclo di vita del SW

- ❑ **Modelli Prescrittivi** → fasi ben codificate in successione, organizzazione e metodi:
  - Modello a cascata (waterfall): fasi strettamente sequenziali
  - Modello a V
  
- ❑ **Modelli Adattivi** → flessibilità, adattabilità, meno documentazione «esterna»:
  - Modello incrementale: realizzazione in più cicli
  - Modello evolutivo: ciclico
  - Modelli Agili: SCRUM, Extreme programming

## Modello a cascata

Modello di sviluppo basato sulla **esecuzione in sequenza di passaggi** da una fase alla successiva in un **ordine strettamente lineare**



## Modello a cascata

- ❑ è storicamente associato alla nascita dell'ingegneria del software
- ❑ E' uno dei **cicli più sicuri** nel raggiungere i risultati e gli obiettivi attesi (tempi, costi, qualità), **quando i requisiti sono definiti e stabili fin dall'inizio**

Il ciclo standard non è rigido nella sua applicazione ma **è possibile personalizzarlo di volta in volta**, sulla base delle caratteristiche del prodotto software da realizzare (per esempio, la dimensione del prodotto, la classe di rischio, ecc.).

**La personalizzazione avviene nella prima fase utile del ciclo** e questo consente di utilizzare il ciclo sia per piccoli progetti sia per progetti medi e grandi.

## Modello a cascata

Il ciclo a cascata prevede le seguenti fasi:

- ☐ Studio di fattibilità
- ☐ Analisi e specifica dei requisiti
- ☐ Progettazione
- ☐ Sviluppo e unit test
- ☐ Integrazione e test di sistema
- ☐ Rilascio
- ☐ Manutenzione

Il ciclo prevede di procedere con una fase dopo l'altra ed è

**necessario completare una fase prima di poter proseguire con la successiva**



## Fase: Studio di fattibilità

### □ Input

- Idea iniziale

### □ Output

- Studio di fattibilità verificato ed approvato

### □ Controlli

- Verifica dello Studio di fattibilità
- Riesame di fine fase

## Fase: Analisi e specifica dei requisiti

### ❑ Input

- Studio di fattibilità

### ❑ Output

- Requisiti del prodotto verificati, validati ed approvati
- Obiettivi di qualità e metriche del progetto verificati, validati ed approvati
- Classificazione del rischio del progetto
- Selezione del ciclo di vita
- Piano di progetto verificato ed approvato
- Specifica dei requisiti verificata ed approvata
- Definizione impiego di fornitori esterni
- Previsione ed impegno costi e risorse verificata ed approvata
- Piano di dettaglio per la fase successiva

### ❑ Controlli di fase

- Verifica e validazione requisiti del prodotto
- Verifiche di progettazione ed approvazione di tutti gli output di fase
- Approvazione Piano di progetto
- Riesame di fine fase

## Fase: Progettazione

### ❑ Input

- Output fase precedente

### ❑ Attività

- Analisi e Progettazione Architettuale
- Pianificazione e progettazione test
- Gestione configurazione
- Monitoraggio e controllo progetto, Aggiornamento Piano di progetto

### ❑ Output

- Documentazione di Analisi e Progettazione approvata
- Piano di Test e Progettazione Test approvati
- Verbalì Verifiche di progettazione su Analisi, Disegno, Piano di Test
- Verbalì Verifica Documentazione
- Piano di progetto aggiornato

### ❑ Controlli di fase

- Verifiche progettazione ed approvazione di tutti gli output di fase: Analisi, Architettura, Piano e progettazione test
- Riesame di fine fase (più eventuali riesami intermedi)

## Fase: Sviluppo e Unit test

### ❑ Input

- Risultati fase precedente

### ❑ Attività

- Codifica e documentazione del codice
- Unit test ed emissione report
- Gestione anomalie e modifiche
- Gestione configurazione
- Verifiche ed emissione report
- Monitoraggio e controllo progetto, Aggiornamento Piano di progetto

### ❑ Output

- Baseline del prodotto
- Report di Unit Test e Verifiche
- Record Anomalie e Modifiche
- Piano di progetto aggiornato
- Verbale Riesame di Fine fase

### ❑ Controlli di fase

- Verifiche progettazione: Test
- Riesame di fine fase (più eventuali riesami intermedi)

## Fase: Test e Validazione

### ❑ Input

- Output fase precedente

### ❑ Attività

- Preparazione documentazione prodotto
- Esecuzione test: funzionale, integrazione, qualificazione, accettazione, etc.
- Gestione anomalie e modifiche
- Gestione configurazione
- Effettuazione Verifiche
- Emissione Report di Test
- Monitoraggio e controllo progetto, Aggiornamento Piano di progetto

### ❑ Output

- Baseline prodotto
- Report di Esecuzione Test
- Record Anomalie e Modifiche
- Piano di progetto aggiornato

### ❑ Controlli di fase

- Verifiche progettazione: Test
- Riesame di fine fase (più eventuali riesami intermedi)

## Fase: Validazione, qualificazione, delivery

### ❑ Input

- Risultati fase precedente

### ❑ Attività

- Test di Validazione ed emissione Report
- Gestione anomalie e modifiche
- Preparazione baseline di rilascio
- Monitoraggio e controllo progetto, Aggiornamento Piano di progetto
- Preparazione Piano di Manutenzione
- Qualificazione prodotto
- Rilascio

### ❑ Output

- Report di Validazione
- Verbale di riesame e qualificazione
- Piano di Commessa, aggiornato oppure Piano di Manutenzione
- Baseline Prodotto rilasciato

### ❑ Controlli di fase

- Validazione
- Riesame finale di qualificazione prodotto e rilascio

## Punti di forza

- ❑ **Flessibilità e personalizzazione** del ciclo di vita:
  - può essere personalizzato e alcune fasi sono sovrapponibili
- ❑ **Gestione dei rischi di progetto (tempi, costi e qualità)** → se i requisiti sono definiti a priori, i rischi vengono minimizzati
- ❑ **Aspetti organizzativi e gerarchici** → permette di chiarire i ruoli e le responsabilità per le diverse fasi ed attività

## Punti di debolezza

- ❑ **Non adatto quando non sono definiti a priori i requisiti del prodotto finale.** Si basa sulla ipotesi che:
  - sia possibile definire esattamente quali sono i requisiti del sistema
  - i requisiti siano stabili→ Difficilmente applicabile nel SW
- ❑ Le verifiche sono previste in fasi avanzate del ciclo di vita:
  - **Ritardata rilevazione di problemi e anomalie**
  - **Maggiori costi e criticità per la correzione**
- ❑ La validazione con il Cliente è prevista in fase avanzata del ciclo di vita:
  - **Disallineamento** tra la realizzazione e l'esigenza
  - Lo sviluppatore procede **senza confronto con il Cliente** se non al termine della attività
  - **Insoddisfazione, contestazioni, maggiori costi e criticità per la correzione**



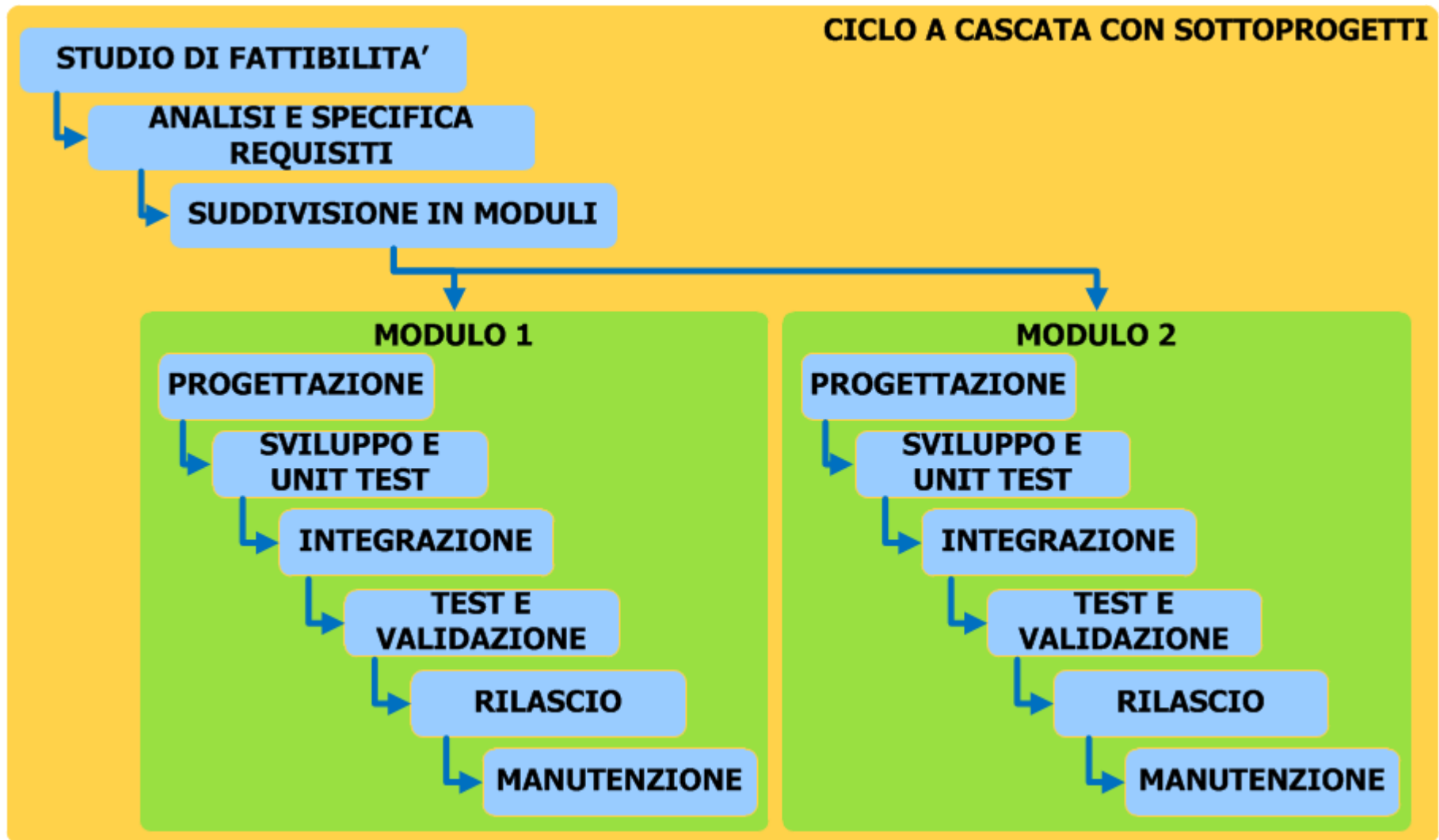
**Personalizzazioni e Varianti del ciclo a cascata** → migliore gestione dei punti di debolezza.

❑ **Ciclo a cascata con fasi sovrapposte:**

- anticipa/sovrappone parzialmente alcune fasi (es.: «progettazione» con «sviluppo e test») → per terminare più rapidamente
- si deve comunque applicare il ciclo completo, prevedendo anche le verifiche e i riesami
- al termine delle fasi sovrapposte si effettua un riesame completo.

❑ **Ciclo a cascata con sottoprogetti :**

- se l'architettura lo consente, è possibile suddividere il progetto in sotto-progetti a cui applicare l'intero ciclo
- richiede un'analisi completa del sistema → definire l'architettura generale e la suddivisione in moduli e componenti il più possibile indipendenti
- le unità logiche vengono sviluppate in parallelo con un processo per fasi



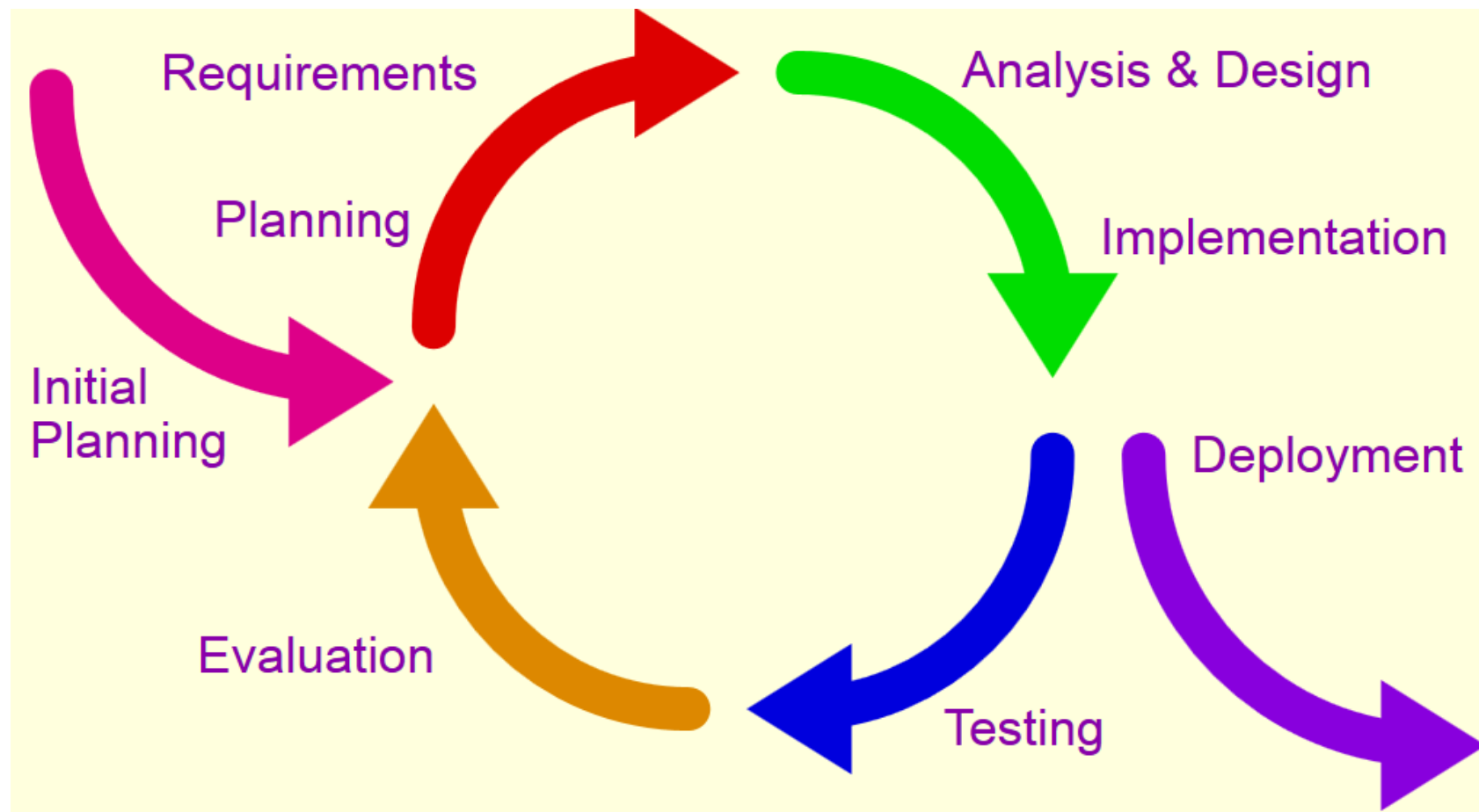
## Modelli evolutivi (incrementali e/o iterativi)

- ❑ Creati per **gestire alcuni punti di debolezza del ciclo a cascata** (necessità di cambiamenti)  
→ **rilevazione tempestiva dei cambiamenti** e **flessibilità nella loro gestione**.

**Ripetizione di più iterazioni del ciclo di sviluppo**, costituito da una sequenza di passi principali:

- ❑ pianificazione
- ❑ analisi dei requisiti
- ❑ progetto
- ❑ implementazione
- ❑ prove
- ❑ valutazione

**Il ciclo può essere ripetuto diverse volte** fino al soddisfacimento dei requisiti



## Modelli evolutivi (incrementale o iterativo)

- ❑ Il **software è prodotto in modo incrementale**, in iterazioni successive
- ❑ **Ogni iterazione produce una parte nuova** oppure una versione **perfezionata** del sistema complessivo.
- ❑ **Al termine di ciascuna iterazione** ne viene **valutato il risultato** e vengono **definite le iterazioni successive**
- ❑ Il processo **termina** quando si raggiunge la **soddisfazione dei requisiti**

## Modello incrementale o modello iterativo

- ❑ Importante avere fin dall'inizio della progettazione, una visione abbastanza chiara dell'intero progetto → organizzare in modo che la realizzazione della generica versione risulti utile per la realizzazione della versione successiva.
- ❑ indicato quando la specifica dei requisiti è difficoltosa e di difficile redazione formale.
- ❑ Supporta la creazione di prototipi, che favoriscono il dialogo con il cliente e la validazione dei requisiti.

## Modello a V

- ❑ Il **V-model** o **Modello a V** è un'estensione del modello a cascata
- ❑ Il modello evidenzia e gestisce la **relazione** tra ogni **fase dello sviluppo** del software e la **relativa fase di test**
- ❑ Evidenzia il concetto di tracciabilità
- ❑ E' un metodo strutturato, in cui ogni fase è implementabile a partire dalla documentazione dettagliata della fase precedente.
- ❑ Le **attività di testing iniziano già all'inizio del progetto** prima della codifica e ciò consente di prevenire e gestire più efficacemente le anomalie e i problemi

## Modello a V

- ❑ Molto utilizzato nella realizzazione di **sistemi safety critical** (sistemi embedded, sistemi medicali, controllo del traffico aereo, ferrovie, avionica, ecc..)
- ❑ Aiuta a garantire **l'affidabilità e la sicurezza** del sistema



