

# **Il Ciclo Vitale dello Sviluppo Software (SDLC) Per Piccole e Medie Applicazioni con Database**

ID Documento: REF-0-02  
Versione: 1.0d





---

## INDICE

---

<b>INTRODUZIONE .....</b>	<b>4</b>
LA CASCATA SDLC .....	4
VARIAZIONI CONSENTITE.....	5
ALTRI MODELLI SDLC .....	6
RIFERIMENTI .....	7
<b>STADIO GENERICO .....</b>	<b>8</b>
AVVIAMENTO.....	8
ITERAZIONE INFORMALE .....	9
ITERAZIONE FORMALE.....	9
ACCERTAMENTO QUALITÀ .....	10
USCITA.....	11
<b>STADI SDLC .....</b>	<b>12</b>
PANORAMICA .....	12
PIANIFICAZIONE.....	13
DEFINIZIONE DEI REQUISITI .....	14
PROGETTAZIONE .....	16
SVILUPPO .....	17
INTEGRAZIONE & TEST.....	18
INSTALLAZIONE & ACCETTAZIONE .....	19
<b>CONCLUSIONE .....</b>	<b>20</b>
LIMITAZIONE DELL'AMBITO.....	20
CRESCITA PROGRESSIVA .....	20
STRUTTURA PRE-DEFINITA.....	21
PIANIFICAZIONE PER INCREMENTI.....	21

---

## INTRODUZIONE

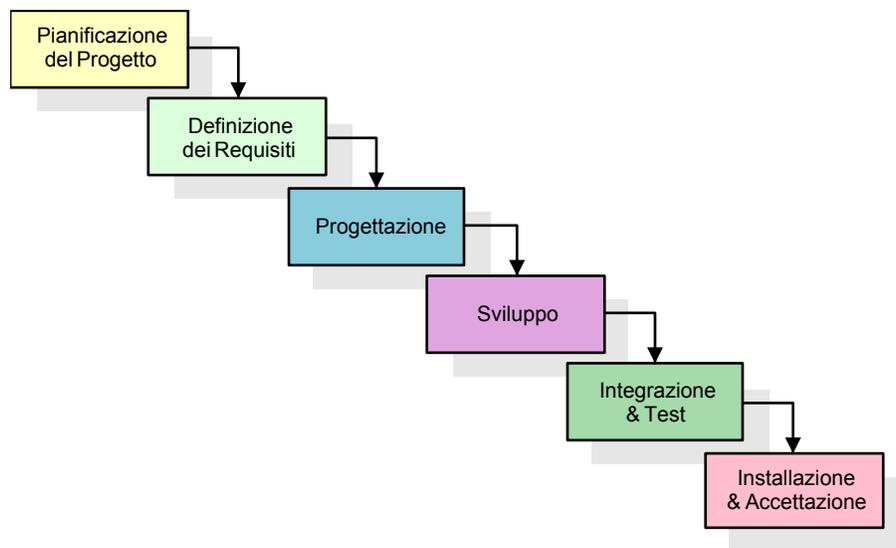
---

Questo documento descrive il Ciclo Vitale dello Sviluppo Software (SDLC – Software Development Life Cycle) per piccole e medie applicazioni con database. Questo capitolo presenta una panoramica dello SDLC insieme con altri possibili modelli di cicli vitali e relativi riferimenti. Il capitolo successivo descrive i processi interni che sono comuni attraverso tutti gli stadi dello SDLC, e il terzo capitolo tratta gli input e output e processi di ciascuno stadio. Infine, la conclusione descrive i 4 concetti che sono alla base di questo SDLC.

### LA CASCATA SDLC

---

I piccoli o medi progetti software con database possono essere suddivisi in sei stadi:



La relazione di ciascuno stadio con gli altri può essere descritta approssimativamente come una cascata, ove gli output provenienti da uno stadio servono come input iniziale per lo stadio successivo.

Durante ciascuno stadio, vengono raccolte o sviluppate altre informazioni che, combinate con l'input ricevuto, sono poi utilizzate per produrre l'output dello

stadio. E' importante notare che le informazioni aggiunte sono di portata limitata: "nuove idee" che porterebbero il progetto in direzioni non anticipate nei requisiti iniziali ad alto livello, non sono incorporate nel progetto. Piuttosto, idee per nuove possibilità o funzionalità che sono fuori dell'ambito attuale, sono conservate per essere poi considerate in un tempo successivo.

Dopo che il progetto è completato, il Primario Rappresentante dello Sviluppo e il Primario Rappresentante Utente, in concerto con altro personale del cliente e del gruppo di sviluppo preparano una lista di raccomandazioni per migliorare e valorizzare il presente software.

## **PROTOTIPI**

Il team di sviluppo, per chiarire dei requisiti o degli elementi di progetto, può generare dei prototipi o schemi di videate, resoconti e processi. Quantunque alcuni prototipi possano apparire completi e rifiniti, essi sono simili alla sceneggiatura di un film: tutto sembra vero dal davanti, ma non vi è nulla alle spalle.

Quando un prototipo è generato, lo sviluppatore crea la quantità minima di codice necessaria per rendere chiari i requisiti o gli elementi progettuali in esame. Non viene fatto alcuno sforzo per aderire a degli standard di creazione del codice, creare una robusta gestione degli errori, o per l'integrazione con le altre tabelle o moduli del database. Di conseguenza, è generalmente più costoso aggiungere a un prototipo i vari elementi necessari per generare un modulo da mettere in produzione, che riscrivere il modulo da zero, utilizzando il documento di progetto finale.

Per queste ragioni, i prototipi non sono mai intesi per un uso in produzione, e sono generalmente menomati in un modo o nell'altro, per impedire che vengano utilizzati per errore come moduli in produzione dagli utenti finali.

## **VARIAZIONI CONSENTITE**

In taluni casi, vengono fornite nuove informazioni al team di sviluppo, informazioni che richiedono cambiamenti nei risultati degli stadi precedenti. In tali casi, lo sviluppo è di solito sospeso fino a che i cambiamenti non sono stati conciliati con il progetto e i nuovi risultati sono passati avanti in cascata, finché il progetto raggiunge il punto in cui era stato sospeso.

Il Rappresentante Utente (PER – Primary End-user Representative) e il Rappresentante Sviluppatore (PDR – Primary Developer Representative) possono, a loro discrezione, consentire che lo sviluppo continui mentre i prodotti resi dagli stadi precedenti vengono aggiornati, nei casi in cui gli impatti siano minimi e di portata rigorosamente limitata. In questi casi, i cambiamenti dovranno essere

accuratamente seguiti, per essere certi che tutte le conseguenze siano trattate appropriatamente.

## **ALTRI MODELLI SDLC**

---

Il modello a cascata è uno dei tre modelli comunemente citati di ciclo vitale. Altri includono il modello a Spirale e il lo Sviluppo Rapido di Applicazioni (RAD – Rapid Application Development), spesso riferito come modello a Prototipi.

### **CICLO VITALE A SPIRALE**

Il modello a spirale comincia con un passo iniziale attraverso il modello standard a cascata, utilizzando un sottoinsieme dei requisiti totali per sviluppare un prototipo ben solido. Dopo un periodo di valutazione, il ciclo inizia nuovamente, aggiungendo nuove funzionalità e rilasciando il prototipo successivo. Questo processo continua, con il prototipo che diventa sempre più grande a ogni iterazione. Da qui la “spirale.” Tale ciclo viene anche detto “Evolutivo.”

La teoria è che l’insieme di funzionalità sia per sua natura gerarchico, con le funzionalità aggiunte che si basano sulle prime. Questo è vero per sistemi in cui l’intero problema è ben definito sin dall’inizio, come nel software di modellazione e simulazione. I progetti di database orientati al commercio non godono di questo vantaggio. Molte delle funzioni in un database sono essenzialmente indipendenti le une dalle altre, sebbene possano utilizzare dei dati comuni. Come risultato, i prototipi soffrono degli stessi difetti presentati dal ciclo vitale a prototipi descritto più avanti. Per questa ragione, il gruppo di sviluppo ha deciso contro l’uso del ciclo vitale a spirale per progetti di database.

### **SVILUPPO RAPIDO DI APPLICAZIONI (RAD) / CICLO VITALE A PROTOTIPI**

Essenzialmente, RAD è l’approccio “prova prima di acquistare” applicato allo sviluppo software. La teoria è che gli utenti finali possano produrre risposte migliori esaminando un sistema in reale, piuttosto che lavorare rigorosamente con la documentazione. I cicli di sviluppo basati sul RAD danno come risultato un minore livello di rigetto dell’applicazione, quando questa è posta in produzione, ma questo successo sovente avviene a spese di un drammatico superamento nei costi e tempi del progetto.

L’approccio RAD è stato reso possibile da significativi avanzamenti degli ambienti di sviluppo software che permettono di generare rapidamente formati e altre caratteristiche dell’interfaccia utente. L’utente finale può lavorare direttamente con i formati finali come se fosse in produzione. Questo lascia poco spazio all’immaginazione e molti errori sono catturati utilizzando questo processo.

L'inconveniente del RAD è la propensione dell'utente finale nell'introdurre deviazioni dall'ambito dello sviluppo. Poiché sembra così facile per lo sviluppatore produrre i formati di base, dovrà essere anche facile aggiungere una o due altre piccole funzionalità. Nella maggior parte dei fallimenti di cicli vitali RAD, gli utenti finali e gli sviluppatori erano catturati in un ciclo senza fine di migliorie, con gli utenti che chiedevano sempre di più e gli sviluppatori che cercavano di soddisfarli. I partecipanti perdevano di vista l'obiettivo di produrre un utile sistema di base in favore di un canto delle sirene di scintillante perfezione.

Per questo motivo, il team di sviluppo software non utilizza un puro approccio RAD, ma piuttosto armonizza un uso limitato dei prototipi nello sviluppo dei requisiti e del progetto, nel corso di un ciclo vitale a cascata convenzionale. I prototipi sviluppati sono specificamente focalizzati su una parte dell'applicazione e non forniscono un'interfaccia integrata. I prototipi sono usati per convalidare i requisiti e gli elementi di progetto, mentre lo sviluppo di altre variazioni all'interfaccia utente o di requisiti aggiuntivi non immediatamente sostenuti dall'ambiente di sviluppo, è attivamente scoraggiato.

## RIFERIMENTI

---

Nello sviluppo di questa descrizione SDLC, sono stati utilizzati come guida gli standard appresso indicati. Questi standard sono stati rivisti e confezionati per adattarsi ai requisiti specifici dei piccoli progetti di database.

- ANSI/IEEE 1028: Standard for Software Reviews and Audits
- ANSI/IEEE 1058.1: Standard for Software Project Management Plans
- ANSI/IEEE 1074: Standard for Software Lifecycle Processes
- SEI/CMM: Software Project Planning Key Process Area

Questo documento fa un ampio uso di terminologia specifica appartenente all'ingegneria del software. Un glossario dei termini correnti è disponibile on-line all'indirizzo seguente:

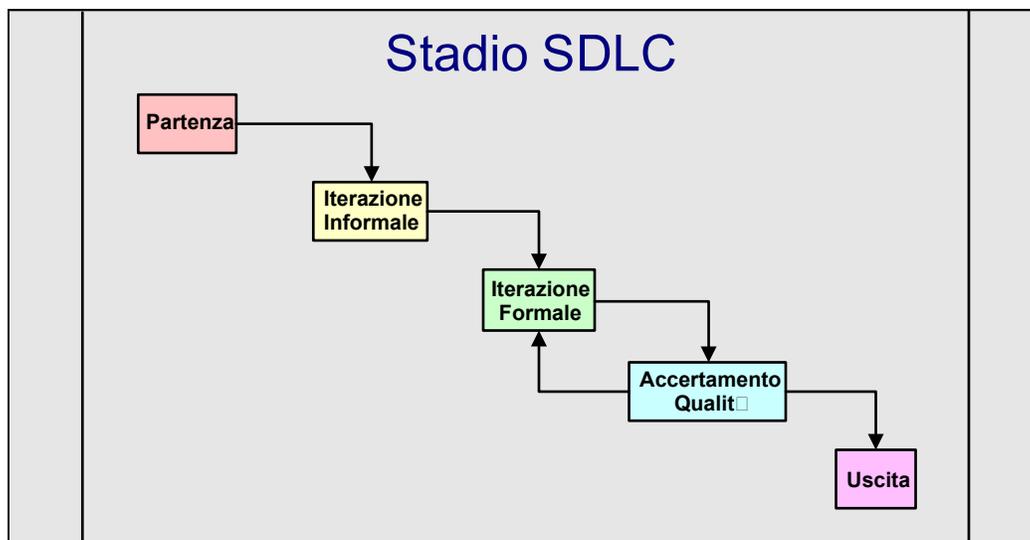
- [www.easystyle.it/refs/seglossary.pdf](http://www.easystyle.it/refs/seglossary.pdf)

---

## STADIO GENERICO

---

Ciascuno stadio del ciclo vitale di sviluppo segue cinque processi interni standard. Questi processi stabiliscono uno schema di comunicazione e documentazione inteso a familiarizzare ogni partecipante con la situazione corrente, e quindi minimizzare i rischi al piano del progetto in corso. Questa descrizione di stadio generico è data per evitare ripetute spiegazioni di questi processi interni in ciascuna delle descrizioni dei successivi stadi del ciclo vitale dello sviluppo software. I cinque processi standard sono: Partenza, Iterazione Informale, Iterazione Formale, Accertamento Qualità e Uscita dallo stadio.



---

### AVVIAMENTO

---

Ciascuno stadio comincia con una riunione di partenza che può essere condotta sia di persona o via tele-conferenza Web. Lo scopo di questa riunione di partenza è rivedere i risultati dello stadio precedente, vedere ogni altro input richiesto dal nuovo stadio, esaminare le attività previste e i risultati richiesti allo stadio corrente, rivedere ogni questione lasciata aperta. Il PDR è responsabile della preparazione dell'agenda e del materiale da presentare a questa riunione. Tutti i partecipanti al progetto sono invitati alla riunione di partenza di ciascuno stadio.

## **ITERAZIONE INFORMALE**

---

La maggior parte del lavoro creativo di ciascuno stadio avviene in questo processo. I partecipanti lavorano insieme per raccogliere informazioni aggiuntive e rifinire gli input dello stadio in bozze consegnabili. Le attività di questo stadio possono includere interviste, incontri, la generazione di prototipi, e corrispondenza elettronica. Tutte queste comunicazioni sono considerate informali e non sono registrate come minute, documenti da registrare, software controllato o memoria ufficiale.

L'intento è quello di incoraggiare, piuttosto che inibire, il processo di comunicazione. Il processo termina quando la maggioranza dei partecipanti è d'accordo che il lavoro è completo e che è tempo di generare delle bozze consegnabili per un controllo formale e commento.

## **ITERAZIONE FORMALE**

---

In questo processo vengono generate delle bozze consegnabili per un controllo formale e commento. Ciascun elemento è stato introdotto durante la partenza dello stadio, ed è inteso a soddisfare uno o più risultati dello stadio corrente. A ciascuna bozza consegnabile è assegnato un numero di versione e posta sotto controllo di configurazione.

I partecipanti che esaminano le bozze di consegna, sono responsabili del riferire al PDR per posta elettronica sia gli errori trovati, sia ogni altra obiezione che possono avere. Il PDR a sua volta consolida questi resoconti in una serie di questioni associate a quella specifica versione del materiale consegnabile. La persona incaricata dello sviluppo del materiale consegnabile lavora per risolvere le questioni riportate e quindi rilascia una nuova versione del consegnabile per un riesame. Questo processo si ripete fino a che tutte le questioni sono risolte per ciascuna parte consegnabile. Non esiste un controllo formale o firma ufficiale per questa parte del processo. L'intento è quello di incoraggiare l'esame, le obiezioni e i commenti.

A discrezione del PDR e del PER, alcune questioni possono essere rimandate per la soluzione in uno stadio successivo del ciclo vitale dello sviluppo. Tali questioni sono dissociate dalle parti consegnabili, e possono essere etichettate come "questioni aperte". Le questioni aperte sono riesaminate durante la riunione di partenza di ciascuno stadio successivo.

Una volta che tutte le questioni che riguardano una parte consegnabile sono state risolte oppure classificate come questioni aperte, si prepara la bozza finale (rilascio) del consegnabile e la si consegna al PDR. Quando tutte le bozze finali di tutti gli elementi richiesti come prodotto dello stadio sono pronte, il PDR rivede

l'insieme del consegnabile, rivede la quantità di lavoro speso in questo stadio del progetto, e usa queste informazioni per aggiornare il piano del progetto.

L'aggiornamento del piano del progetto include una lista dettagliata dei compiti, la loro programmazione nel tempo, e stima del lavoro per lo stadio successivo. Gli stadi oltre al successivo (meta-stadi) nel piano del progetto sono aggiornati per includere una stima di massima di tempo e sforzo previsti, basandosi sull'esperienza corrente.

I meta-stadi sono indicati solo in linea di massima nel piano del progetto, e sono inclusi soprattutto per scopi informativi; l'esperienza diretta ha mostrato che è molto difficile pianificare dettagliatamente i compiti e le attività dei meta stadi in un ciclo di vita dello sviluppo software. L'aggiornamento del piano del progetto e la programmazione nel tempo è una parte consegnabile o resoconto standard per ciascuno stadio del progetto. Il PDR fa allora circolare il piano del progetto aggiornato e la programmazione nel tempo per controllo e commento, e ripete questi documenti fino a quando tutte le questioni sono state risolte ovvero riclassificate come aperte.

Una volta che il piano del progetto e programmazione sono ultimati, tutte le parti consegnabili finali dello stadio sono rese disponibili a tutti i partecipanti, e il PDR avvia il processo successivo.

## **ACCERTAMENTO QUALITÀ**

---

Questo è il processo d'accertamento formale della qualità per ciascuno stadio. In un piccolo progetto di sviluppo software, le parti consegnabili di ciascuno stadio sono generalmente piccole talché non è conveniente verificarne la congruenza con standard di qualità finché dette parti non sono state pienamente sviluppate. Per cui è previsto un singolo accertamento per ciascuno stadio.

Questo processo inizia quando il PDR stabilisce un accertamento con il Revisore Assicurazione Qualità (QAR – Quality Assurance Reviewer) indipendente, un Utente finale selezionato (solitamente un Esperto della Materia) e un Tecnico Revisore selezionato.

Questi revisori verificano formalmente ciascuna parte consegnabile per emettere giudizi sulla qualità e sulla validità del lavoro, come anche sulla conformità agli standard definiti per i consegnabili di quella classe. Gli standard di qualità per i prodotti consegnabili sono definiti nella sezione assicurazione qualità del software del piano del progetto.

L'Utente finale Revisore ha il compito di verificare la completezza e accuratezza del consegnabile in termini delle funzionalità desiderate dal software. Il Revisore Tecnico determina se il consegnabile contiene informazioni tecniche complete e accurate.

Il QAR ha solamente il compito di verificare la completezza e la congruità del consegnabile a fronte degli standard associati alla classe del consegnabile. Il QAR può fare delle raccomandazioni, ma non può sollevare delle questioni formali che non siano in relazione con lo standard del consegnabile.

Ciascun revisore segue una lista formale durante il controllo, indicando il livello d'aderenza con ciascun elemento della lista. Il riferimento è il piano d'assicurazione qualità per questo progetto per gli standard della classe dei consegnabili e lista di controllo associata.

Ogni questione sollevata di revisori riguardante uno specifico prodotto consegnabile viene annotata e data al personale responsabile per la generazione di quel consegnabile. Il consegnabile revisionato è quindi rilasciato ai partecipanti del progetto per un altro processo d'iterazione formale. Una volta che tutte le questioni per quel consegnabile saranno state risolte, il consegnabile sarà nuovamente sottoposto ad accertamento di qualità. Quando tutti i tre revisori avranno indicato conformità di quel consegnabile agli standard, il PDR rilascerà un resoconto finale d'accertamento qualità e inizierà il processo successivo.

## **USCITA DALLO STADIO**

---

L'uscita dallo stadio è il veicolo per assicurare il consenso dei partecipanti principali al progetto a proseguire e procedere nello stadio successivo dello sviluppo. Lo scopo del processo d'uscita dallo stadio è consentire a tutto il personale coinvolto di rivedere il piano corrente del progetto e i risultati dello stadio, avere un luogo dove sollevare questioni o riserve e assicurare che esiste un piano d'azione accettabile per tutte le questioni lasciate aperte.

Il processo inizia quando il PDR avvisa tutti i partecipanti che i consegnabili dello stadio corrente sono stati portati a termine e approvati attraverso il resoconto d'Accertamento. Il PDR quindi programma un riesame d'uscita dallo stadio con lo sponsor del progetto e con il PER come minimo. Tutti gli altri partecipanti interessati sono liberi di partecipare. Questo incontro può essere condotto di persona oppure in tele-conferenza Web.

Il processo d'uscita dallo stadio termina con la ricevuta di consenso, da parte di chi deve approvare, a procedere allo stadio successivo. Questo è fatto generalmente registrando le minute del controllo d'uscita come documenti ufficiali, sia con una firma reale che con una firma virtuale dello sponsor del progetto, del PER e del PDR.

---

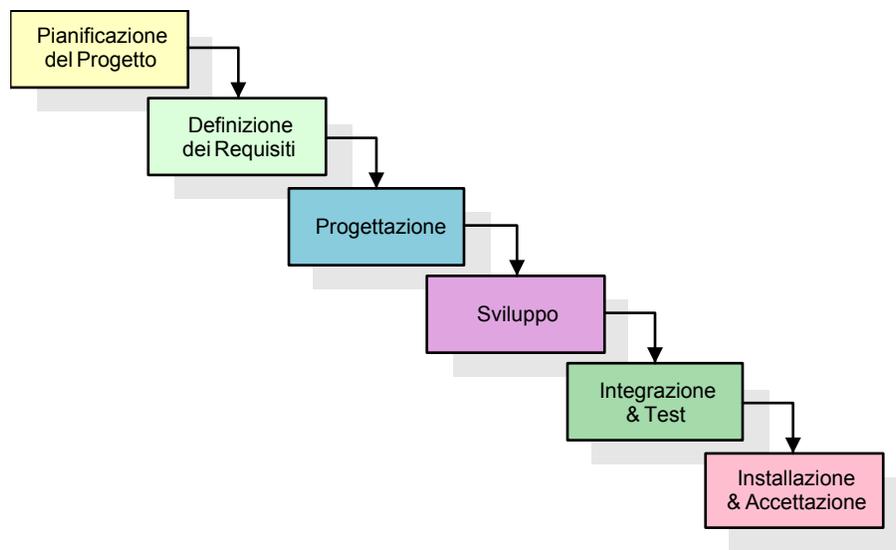
## STADI SDLC

---

### PANORAMICA

---

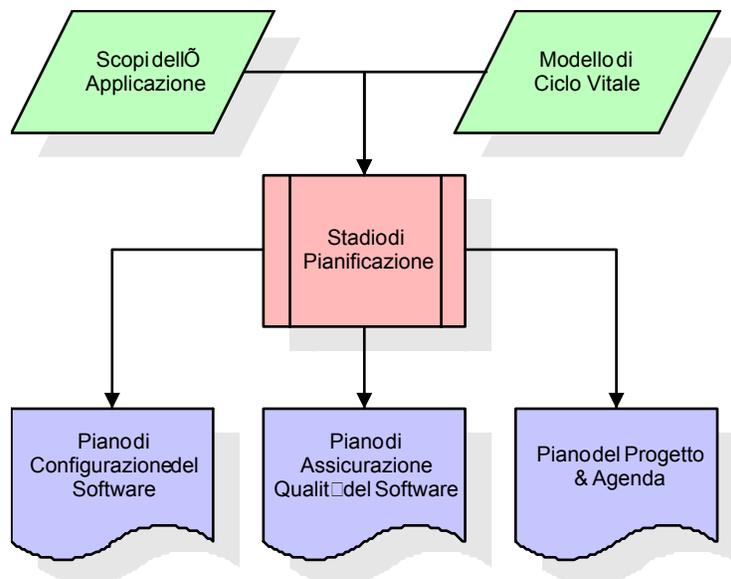
I sei stadi dello SDLC sono progettati per costruire uno sull'altro, prendendo i risultati dello stadio precedente, aggiungendo altro lavoro, e producendo altri risultati che moltiplicano lo sforzo precedente e sono direttamente attribuibili agli stadi precedenti. Questo approccio dall'alto verso il basso è inteso a generare un prodotto di qualità che soddisfi le intenzioni originali del cliente.



Troppi lavori di sviluppo software vanno a monte quando il team di sviluppo e il personale del cliente vengono intrappolati nelle possibilità di automazione. Invece di focalizzarsi sulle caratteristiche prioritarie, il team si può impantanare in un mare di funzionalità “carine da avere” che non sono essenziali alla soluzione del problema, ma che in sé stesse sono molto attraenti. Questa è la causa di una larga percentuale di lavori di sviluppo falliti o abbandonati, e questa è la ragione principale per cui il team di sviluppo utilizza l’SDLC a Cascata.

## PIANIFICAZIONE

Lo stadio di pianificazione stabilisce una visione di insieme del prodotto software proposto e usa questa per stabilire la struttura base del progetto, valutare fattibilità e rischi associati al progetto, e descrivere degli approcci organizzativi e tecnici appropriati.



La sezione più critica del piano del progetto è un elenco dei requisiti ad alto livello del prodotto, anche riferiti come scopi. Tutti i requisiti del prodotto software, da sviluppare durante lo stadio di definizione dei requisiti, fluiscono da uno o più di questi scopi. L'informazione minima per ciascuno scopo consiste di un titolo e di una descrizione testuale, anche se possono essere incluse delle ulteriori informazioni e riferimenti a documenti esterni.

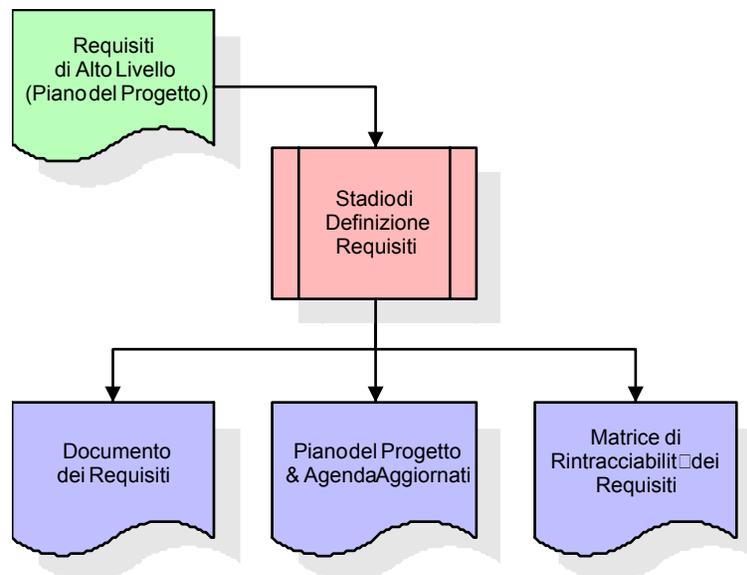
I prodotti dello stadio di pianificazione del progetto sono il piano di configurazione del software, il piano di assicurazione della qualità e un piano del progetto e programma, con un elenco dettagliato delle attività del prossimo stadio dei Requisiti e una stima di massima dei lavori per i successivi meta-stadi.

## DEFINIZIONE DEI REQUISITI

Il processo di raccolta dei requisiti prende come dati gli scopi identificati nella sezione dei requisiti ad alto livello del piano del progetto. Ogni obiettivo viene raffinato in una serie di uno o più requisiti.

Questi requisiti definiscono le principali funzionalità dell'applicazione prefissa, definiscono le aree dati operazionali e di riferimento, e definiscono le entità iniziali dei dati. Le principali funzionalità includono la gestione di processi critici, così come inserimenti, uscite e resoconti d'importanza cruciale. Una gerarchia di classi di utente è sviluppata ed associata a queste funzionalità principali, aree dati ed entità dei dati.

Ciascuna di queste definizioni è chiamata un Requisito. I Requisiti sono contrassegnati da un identificatore univoco e, come minimo, contengono un titolo del requisito e una descrizione testuale.



Questi requisiti sono pienamente descritti nei prodotti principali di questo stadio: il Documento dei Requisiti e la Matrice di Rintracciabilità dei Requisiti (RTM - Requirements Traceability Matrix). Il documento dei requisiti contiene una descrizione completa di ciascun requisito, inclusi diagrammi e riferimenti a documenti esterni, se necessario. È da notare che una lista dettagliata delle tabelle e dei campi del database *non* è inclusa nel documento dei requisiti.

Il titolo di ciascun requisito è anche inserito nella prima versione della RTM, insieme al titolo di ciascun obiettivo preso dal piano del progetto. Lo scopo della RTM è mostrare che i componenti del prodotto sviluppati durante ciascuno stadio

del ciclo vitale dello sviluppo software, sono connessi formalmente ai componenti sviluppati negli stadi precedenti.

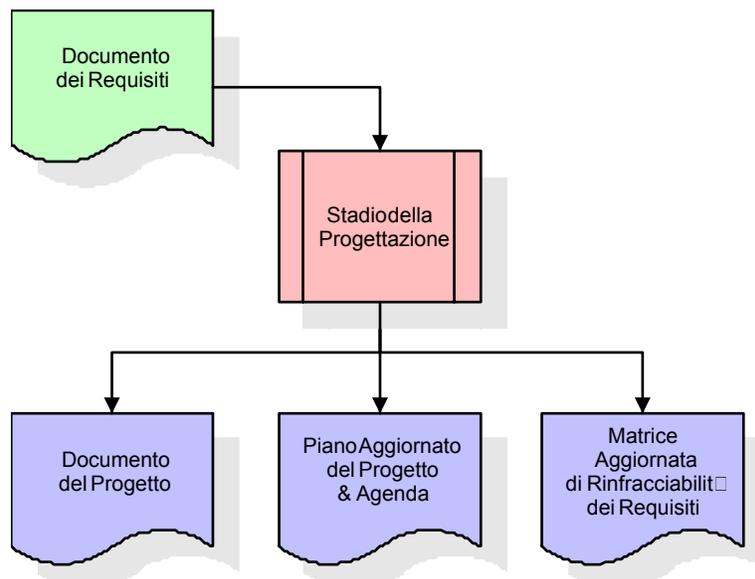
Nello stadio dei requisiti, la RTM consiste in una lista di requisiti ad alto livello o obiettivi, elencati per nome, con una lista dei requisiti associati a ciascun obiettivo, sempre elencati per nome. In questa lista gerarchica, la RTM mostra che ciascun requisito sviluppato durante questo stadio è formalmente connesso a uno specifico obiettivo del prodotto. In questo modo, ciascun requisito può essere attribuito a uno specifico obiettivo del prodotto, di qui il termine di *rintracciabilità dei requisiti*.

I prodotti dello stadio di definizione dei requisiti includono il documento dei requisiti, la RTM, e un piano del progetto aggiornato.

## PROGETTAZIONE

Lo stadio della progettazione prende come dati iniziali i requisiti indicati nel documento dei requisiti approvato. Per ciascun requisito, un insieme di uno o più elementi di progetto viene generato come risultato di interviste, laboratori e/o lavori con prototipi.

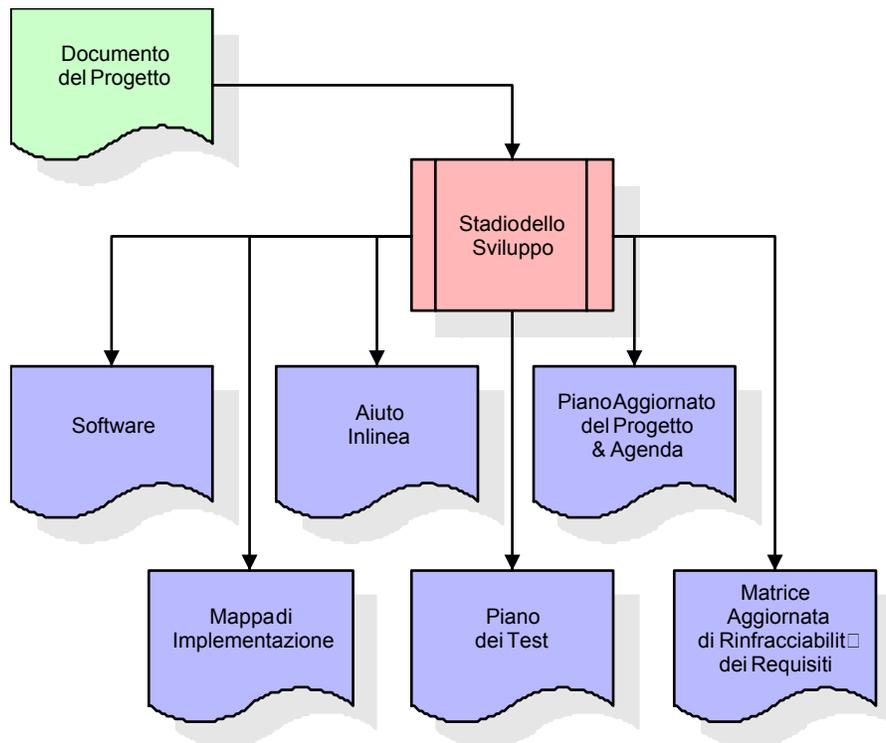
Gli elementi di progetto descrivono in dettaglio le funzionalità desiderate nel software, e generalmente includono schemi gerarchici di funzioni, schemi di formati a video, tabelle di regole di impresa, schemi di processi di impresa, pseudocodice, e uno schema di tutte le relazioni fra entità con un completo dizionario dei dati. Questi elementi di progetto hanno lo scopo di descrivere il software con dettaglio sufficiente affinché un abile programmatore possa sviluppare il software con un minimo d'informazioni supplementari.



Quando il documento del progetto è terminato e accettato, la RTM è aggiornata per mostrare come ciascun elemento di progetto è formalmente associato a uno specifico requisito. I prodotti dello stadi della progettazione sono il documento del progetto, una RTM aggiornata e un piano aggiornato del progetto.

## SVILUPPO

Lo stadio dello sviluppo prende come principali dati iniziali gli elementi del progetto descritti nel documento del progetto approvato. Per ciascun elemento di progetto, verrà prodotto un insieme di uno o più manufatti software. I manufatti software includono, ma non sono limitati a, menù, finestre di dialogo, formati per la gestione dei dati, formati per resoconti, procedure funzioni specializzate. Saranno preparati dei test per ciascun insieme di funzionalità relative ai manufatti software, e sarà sviluppato un aiuto in linea per guidare gli utenti nell'interazione con il software.



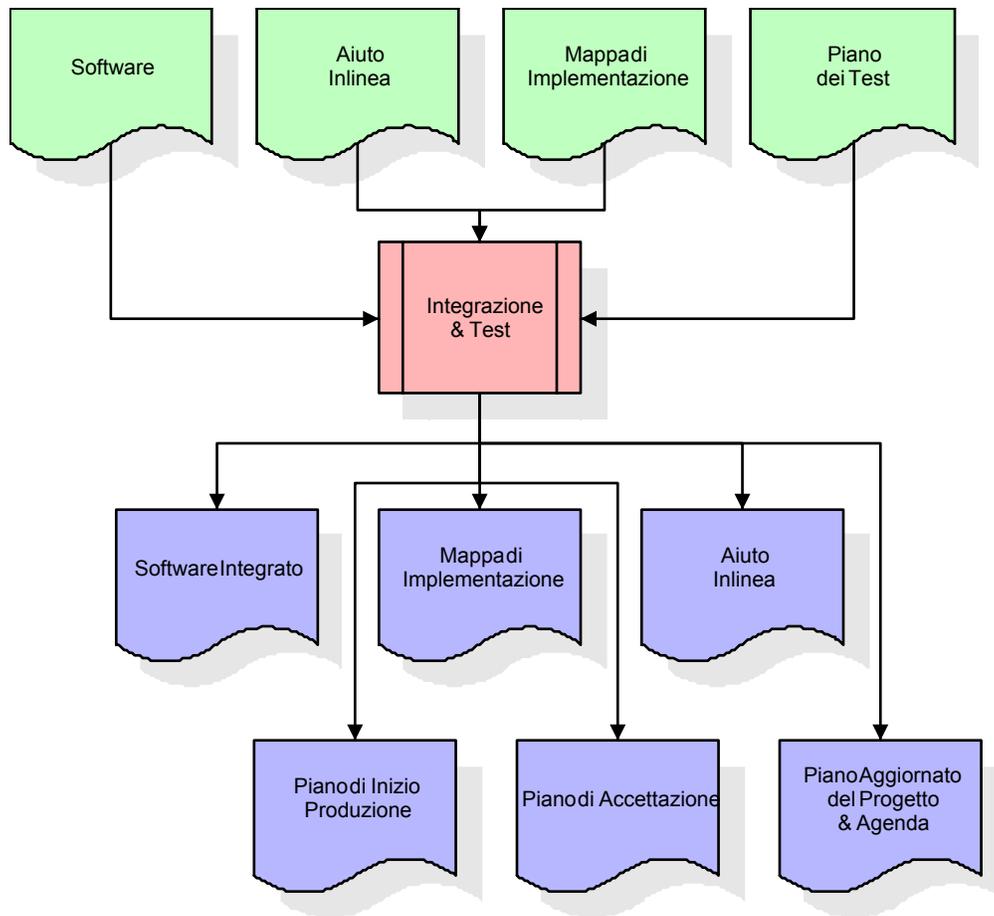
La RTM sarà aggiornata per mostrare che ciascun manufatto sviluppato è collegato a uno specifico elemento di progetto, e che ciascun manufatto sviluppato ha uno o più test di casi d'uso corrispondenti. A questo punto, la RTM è nella sua configurazione finale.

I prodotti dello stadio di sviluppo includono un insieme di software pienamente funzionale che soddisfa i requisiti e gli elementi di progetto documentati precedentemente, un sistema di aiuto in linea che descrive le operazioni del software, una mappa di implementazione che identifica i punti di ingresso primari del codice per tutte le funzioni principali del sistema, un piano dei test che descrive i test da usare per convalidare la correttezza e la completezza del software, una RTM aggiornata e un piano di progetto aggiornato.

## INTEGRAZIONE & TEST

Durante lo stadio di integrazione e test, i manufatti software, l'aiuto in linea e i dati di test sono portati dall'ambiente di sviluppo in un ambiente separato di test. A questo punto, tutti i casi test vengono lanciati per verificare la correttezza e la completezza del software. Il successo nell'esecuzione di un insieme di test conferma un capacità di migrazione completa e robusta.

Durante questo stadio, i dati di riferimento sono finalizzati alla produzione e vengono identificati gli utenti di produzione e collegati ai ruoli a loro appropriati. I dati di riferimento finali (o collegamenti ai documenti sorgente di riferimento) e la lista degli utenti di produzione sono incorporati nel Piano di Avvio Produzione.

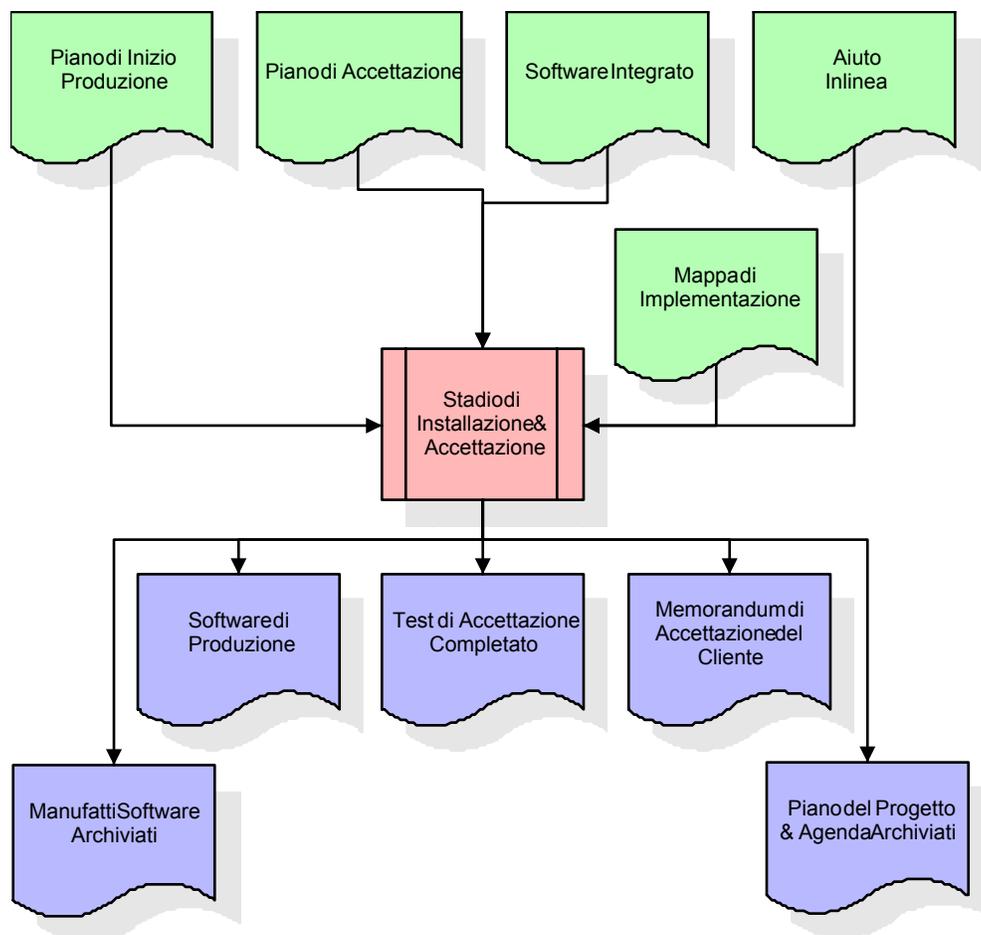


I prodotti dello stadio d'integrazione e test includono un insieme integrato di software, un sistema di aiuto in linea, una mappa di implementazione, un piano di inizio produzione che descrive i dati di riferimento e gli utenti di produzione, un piano di accettazione che contiene l'elenco dei casi dei test, e un piano aggiornato del progetto.

## INSTALLAZIONE & ACCETTAZIONE

Durante lo stadio d'installazione e accettazione, i manufatti software, l'aiuto in linea e i dati iniziali di produzione sono caricati sul server di produzione. A questo punto, tutti i casi test sono lanciati per verificare la correttezza e la completezza del software. L'esecuzione con successo dei test è un prerequisito all'accettazione del software da parte del cliente.

Dopo che il personale del cliente ha verificato che i dati iniziali di produzione sono corretti e che l'insieme dei test sono stati eseguiti con risultati soddisfacenti, il cliente accetta formalmente la consegna del software.



I prodotti principali dello stadio d'installazione e accettazione includono una applicazione in produzione, un insieme di test d'accettazione completati, e un memorandum del cliente di accettazione del software. Infine il PDR inserisce gli ultimi dati di lavoro nell'agenda e blocca il tutto come dato permanente del progetto. A questo punto il PDR "blocca" il progetto archiviando per il futuro tutti gli elementi di progetto, mappe di implementazione, codice e documentazione.

---

## CONCLUSIONE

---

La struttura imposta da questo SDLC è progettata specificamente per massimizzare le probabilità di successo di un lavoro di sviluppo software. A tale fine, lo SDLC si poggia su quattro concetti fondamentali:

- Limitazione dell'Ambito
- Crescita Progressiva
- Struttura Pre-definita
- Pianificazione per Incrementi

Questi quattro concetti si combinano per mitigare i rischi più comuni associati ai lavori di sviluppo software.

### LIMITAZIONE DELL'AMBITO

---

L'ambito del progetto è stabilito dal contenuto dei requisiti di alto livello, chiamati anche obiettivi, che sono incorporati nel piano del progetto. Questi obiettivi sono successivamente precisati o raffinati in requisiti, poi come elementi di progetto e infine come manufatti software.

Questa gerarchia di obiettivi, requisiti, elementi e manufatti è documentata in una Matrice di Rintracciabilità dei Requisiti (RTM). La RTM serve come elemento di controllo per restringere o limitare il progetto all'ambito definito inizialmente.

I partecipanti al progetto sono vincolati a trattare solo quei requisiti, elementi e manufatti che sono direttamente collegabili agli obiettivi del prodotto. Questo previene l'insinuarsi di deviazioni dall'ambito del progetto, che è una delle maggiori cause di fallimento nei progetti software.

### CRESCITA PROGRESSIVA

---

Ciascuno stadio dello SDLC prende i risultati dello stadio precedente come suoi dati iniziali. Altre informazioni sono quindi raccolte o elaborate, utilizzando metodi specifici per ciascuno stadio. Come risultato, i prodotti dello stadio precedente vengono progressivamente accresciuti con informazioni aggiunte.

Con lo stabilire un percorso di accrescere il lavoro precedente, il progetto preclude l'inserimento d'altri requisiti negli stadi successivi. I nuovi requisiti sono messi da parte formalmente dal team di sviluppo per un riferimento successivo, piuttosto che passare attraverso lo sforzo di inserire i nuovi requisiti negli stadi precedenti e riconciliare gli effetti delle aggiunte. Perciò i partecipanti al progetto mantengono a fuoco gli obiettivi originali del prodotto, minimizzando la possibilità di deviare, mostrando una preferenza nel differire le migliorie fuori dalla portata del progetto, piuttosto che incorporarle nel lavoro corrente.

## **STRUTTURA PRE-DEFINITA**

---

Ciascuno stadio ha un insieme pre-definito di processi standard, come l'Iterazione Informale e l'Accertamento Qualità. I partecipanti al progetto si abituano presto a questo modus operandi ripetitivo nel progredire da stadio a stadio. Nella sostanza, questi processi stabiliscono un ritmo comune, o cultura, per il progetto.

Questa struttura pre-definita per ciascuno stadio permette ai partecipanti al progetto di lavorare in un ambiente familiare, dove loro sanno cosa è successo in passato, cosa sta accadendo adesso, e avere aspettative accurate su cosa avverrà nel prossimo futuro. Questo genera un alto livello di comfort, che a sua volta genera un alto livello di cooperazione fra i partecipanti. I partecipanti tendono a fornire le informazioni necessarie e i commenti con regolarità, con meno comunicazioni mancate. Questa consuetudine alla regolarità dei responsi e qualità del livello di comunicazione diventa abbastanza prevedibile, aumentando l'abilità del PDR di prevedere il livello del lavoro negli stadi futuri.

## **PIANIFICAZIONE PER INCREMENTI**

---

L'intero intento della pianificazione incrementale è minimizzare le sorprese, aumentare l'accuratezza, fornire avviso di deviazioni significative dal piano prima possibile nello SDLC, e coordinare le previsioni del progetto con le più recenti informazioni disponibili.

In questo SDLC, il lavoro del piano del progetto è limitato al raccogliere le misure sullo stadio attuale, pianificare lo stadio successivo in dettaglio e limitare la pianificazione degli altri stadi, chiamati anche Meta stadi, ad un livello di massima. Il piano del progetto è aggiornato alla fine di ciascuno stadio; i costi attuali e l'agenda aggiornata vengono combinati con stime accurate delle attività e quantità di lavoro per il prossimo stadio.

Le attività e i compiti dello stadio successivo sono definiti solamente dopo che i prodotti dello stadio corrente sono completati e le misure attuali sono disponibili. Questo consente di fare un piano molto accurato per il prossimo stadio.

L'esperienza diretta ha dimostrato che è estremamente difficile generare qualcosa di più che una stima grossolana della sforzo previsto per i meta stadi.

Le stime per i meta stadi sono incluse solo per permettere una stima grossolana dell'intero progetto e agenda. Le stime per i meta stadi sono riesaminate e riverificate all'uscita di ciascuno stadio. In questo modo, la stima del progetto totale diviene a mano a mano più accurata nel tempo.

All'uscita di ciascuno stadio viene presentato al cliente un piano aggiornato del progetto e agenda. Il cliente è informato dei progressi del progetto, costo e agenda e le misure attuali vengono confrontate con le stime. Questo dà al cliente l'opportunità di confermare che il progetto e sui binari, oppure prendere le misure appropriate se è necessaria una correzione. Il cliente non è mai lasciato "al buio" circa il progredire del progetto.