


# Il ciclo di vita del software



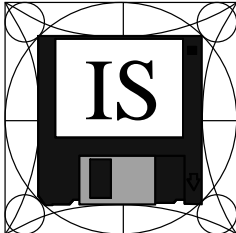
Il ciclo di vita del software  
**Contenuti**

## Il ciclo di vita del software


IS 2001-2008  
Corso di Ingegneria del Software

V. Ambriola, G.A. Cignoni,  
C. Montangero, L. Semini

Con aggiornamenti di: T. Vardanega (UniPD)



Dipartimento di Informatica, Università di Pisa 1/25



Il ciclo di vita del software  
**Contenuti**

- Il ciclo di vita del software
- Il modello a spirale
- Il modello a cascata
- I modelli iterativi
- Seminario: i cicli di vita in ISO/IEC 12207


Dipartimento di Informatica, Università di Pisa 2/25



Il ciclo di vita del software  
**Il concetto di ciclo di vita**

- Concezione → sviluppo → utilizzo → ritiro
- Identificazione delle attività
  - Modelli generici, indipendenti dal prodotto
  - Decomposizione delle fasi in attività (ricerca di coesione)
  - Adozione di terminologia consistente
- Organizzazione delle attività
  - Ordinamento delle attività
  - Definizione di criteri di completamento e di avanzamento


Dipartimento di Informatica, Università di Pisa 3/25



Il ciclo di vita del software  
**Obiettivi e limiti**

- Propedeutico al processo definito
- Richiede controllo di qualità
  - Definizione del processo produttivo (standard aziendale)
  - Identificazione di attività, obiettivi, dipendenze
- Agevola la pianificazione e la gestione dei progetti
- Indipendente da metodi e strumenti di sviluppo
  - Problematica essenziale (invece che accidentale)


Dipartimento di Informatica, Università di Pisa 4/25



Il ciclo di vita del software  
**Evoluzione dei modelli**

- Code-'n-Fix** : un modello in negativo
  - Attività eseguite senza organizzazione
  - Risulta in progetti caotici non gestiti né gestibili
- Modelli organizzati**
  - *Cascata* a rigide fasi sequenziali oppure sue varianti con prototipi e ritorni
  - *Incrementale* realizzazione in più passi
  - *Evolutivo* modelli ciclici con ripetute iterazioni interne
  - *A spirale* contesto allargato e modello astratto

Dipartimento di Informatica, Università di Pisa 5/25

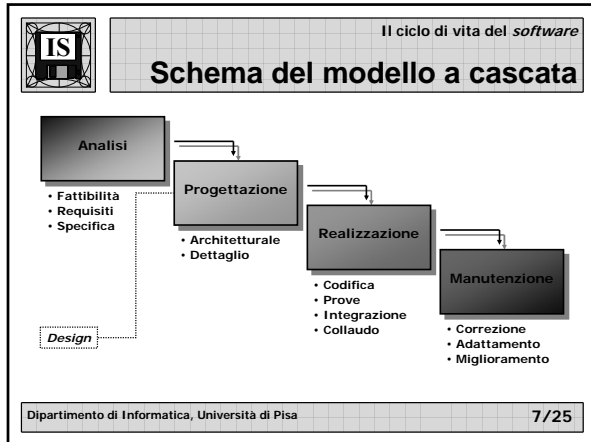


Il ciclo di vita del software  
**(1) Il modello a cascata**

- Definito nel 1970 da Winston W. Royce
  - "Managing the development of large software systems: concepts and techniques"
- Successione di fasi rigidamente sequenziali
  - Non consente ritorno a fasi precedenti
  - Eventi eccezionali fanno ripartire dall'inizio
- Documentazione
  - Ogni fase produce "documenti" che la concretizzano
  - Emissione e approvazione di documenti sono condizione necessaria per l'avvio della fase successiva
    - Modello "document-driven"

Dipartimento di Informatica, Università di Pisa 6/25

# Il ciclo di vita del software

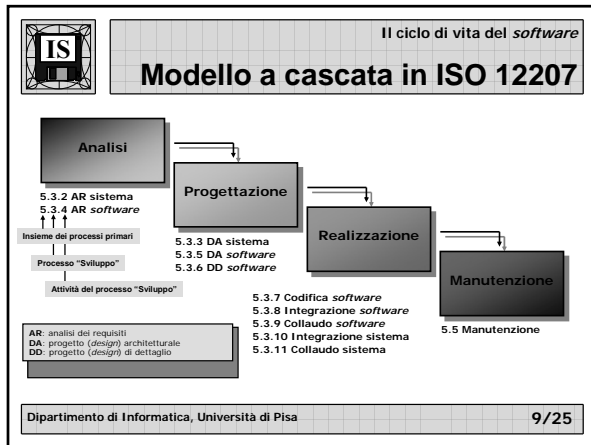


Il ciclo di vita del software

## Caratteristiche delle fasi

- ❑ Descritte in termini di
  - Attività previste e prodotti attesi
  - Contenuti e struttura dei documenti
  - Responsabilità e ruoli coinvolti
  - Scadenze di consegna dei documenti
- ❑ Presentano dipendenze causali e temporali
- ❑ Ciascuna fase prevede attività specifiche

Dipartimento di Informatica, Università di Pisa 8/25



Il ciclo di vita del software

## Varianti del modello a cascata

- ❑ Difetto principale: eccessiva rigidità
  - Stretta sequenzialità tra fasi
  - Non ammette modifiche nei requisiti in corso d'opera
  - Richiede molta manutenzione
  - Esprime una visione burocratica e poco realistica
- ❑ Variante 1: cascata con prototipazione
  - Prototipo "usa e getta"
    - Solo per capire meglio i requisiti
- ❑ Variante 2: cascata con ritorni
  - Entrambe le varianti raggruppano sottosequenze di fasi

Dipartimento di Informatica, Università di Pisa 10/25

Il ciclo di vita del software

## Modelli iterativi

- ❑ Applicabili a qualunque modello di ciclo di vita
- ❑ Consentono maggior capacità di adattamento
  - Evoluzione di problemi, soluzioni possibili e tecnologie utilizzabili
  - Diversificazione dei requisiti del committente
  - I requisiti restano stabili solo in casi molto rari
- ❑ Soluzione generale
  - Decomporre la realizzazione del sistema
  - Identificare e gestire le componenti più critiche
    - Quelle più complesse
    - Oppure quelle i cui requisiti non sono sufficientemente chiari
- ❑ Le iterazioni devono essere pianificate

Dipartimento di Informatica, Università di Pisa 11/25

Il ciclo di vita del software

## Vantaggi dei modelli iterativi

- ❑ Possono produrre "valore" a ogni iterazione
  - Un insieme non vuoto di funzionalità è reso presto disponibile al cliente
- ❑ Le prime iterazioni viste come fasi di prototipazione
  - Che aiutano a fissare meglio i requisiti per le iterazioni successive
- ❑ Aiutano a diminuire il rischio di fallimento
  - Ma senza azzerarlo a causa dei costi aggiuntivi di ciascuna iterazione
- ❑ Se le componenti di sistema essenziali sono sviluppate nelle prime iterazioni
  - Attraversano più fasi di verifica
  - E quindi diventano più stabili con ciascuna iterazione

Tratto da: Ian Sommerville, Software Engineering, 8<sup>a</sup> ed.

Dipartimento di Informatica, Università di Pisa 12/25

# Il ciclo di vita del software

Il ciclo di vita del *software*

## (2) Modello incrementale

- **Analisi e progettazione non sono ripetute**
  - I requisiti sono identificati e fissati completamente
  - L'architettura del sistema è identificata e fissata definitivamente
  - I passi della realizzazione incrementale sono pianificati
    - Per assicurare convergenza entro tempo e costi previsti
- **La realizzazione è incrementale**
  - Progettazione di dettaglio, codifica e prove
  - Integrazione e collaudo
- **Accettazione e rilascio di versioni parziali**
  - Dai requisiti essenziali (prima) a quelli desiderabili (poi)

Dipartimento di Informatica, Università di Pisa 13/25

Il ciclo di vita del *software*

## Modello incrementale in ISO 12207

Dipartimento di Informatica, Università di Pisa 14/25

Il ciclo di vita del *software*

## (3) Modello evolutivo

- **Analisi preliminare**
  - Per identificare i requisiti di massima
  - Per definire l'architettura di massima
  - Per pianificare i passi di analisi e realizzazione evolutiva
- **Analisi e realizzazione di evoluzione**
  - Per raffinamento ed estensione dell'analisi
  - Per progettazione, codifica, prove e integrazione
- **Rilascio di prototipi e poi accettazione finale**

Dipartimento di Informatica, Università di Pisa 15/25

Il ciclo di vita del *software*

## Modello evolutivo in ISO 12207

Dipartimento di Informatica, Università di Pisa 16/25

Il ciclo di vita del *software*

## (4) Il modello a spirale

- **Proposto da Barry W. Boehm nel 1988**
- **Prevede quattro attività principali**
  - Definizione degli obiettivi
  - Analisi dei rischi
  - Sviluppo e validazione
  - Pianificazione
- **È un modello astratto che va specializzato**

Dipartimento di Informatica, Università di Pisa 17/25

Il ciclo di vita del *software*

## Fasi del modello a spirale

- **Definizione degli obiettivi**
  - Requisiti, rischi, piano di gestione
- **Analisi dei rischi**
  - Studio delle conseguenze
  - Valutazione delle alternative
  - con l'ausilio di prototipi e simulazioni
- **Sviluppo e validazione**
  - Realizzazione del prodotto
- **Pianificazione**
  - Decisione circa il proseguimento
  - Pianificazione del ciclo di vita successivo

Dipartimento di Informatica, Università di Pisa 18/25

# Il ciclo di vita del software

Il ciclo di vita del *software*

## Schema del modello a spirale

Dipartimento di Informatica, Università di Pisa 19/25

Il ciclo di vita del *software*

## Aspetti del modello a spirale

- ❑ **Pone grande attenzione sugli aspetti gestionali**
  - Pianificazione delle fasi
  - Analisi dei rischi (modello "risk-driven")
- ❑ **Richiede forte interazione tra committente e fornitore**
  - **Committente:** definizione degli obiettivi  
definizione dei vincoli sulla pianificazione
  - **Fornitore:** sviluppo e validazione
  - **Entrambi:** analisi dei rischi
- ❑ **Applicabile ai modelli tradizionali**

Dipartimento di Informatica, Università di Pisa 20/25

Il ciclo di vita del *software*

## Altri modelli

- ❑ **Extreme Programming (XP)**  
<http://www.extremeprogramming.org>
  - Facilita il coinvolgimento (informale) del committente
    - Non solo nell'analisi, ma anche in progettazione e sviluppo (!)
    - Fortemente incrementale (a passo piccolo)
  - Tende (colpevolmente) a perdere rigore
- ❑ **Component-based**
  - Massima attenzione al riuso sistematico di componenti preesistenti proprie oppure "off-the-shelf"

Dipartimento di Informatica, Università di Pisa 21/25

Il ciclo di vita del *software*

## Modello "component-based"

Dipartimento di Informatica, Università di Pisa 22/25

Il ciclo di vita del *software*

## Ripartizione dei costi sui modelli

- ❑ **Applicazioni serie "normali"**
  - ~ 60% dei costi allo sviluppo
  - ~ 40% alla qualifica
- ❑ **I costi complessivi variano al variare del dominio e del tipo di sistema**
- ❑ **La ripartizione dei costi sulle fasi varia al variare del modello e del dominio**
  - Sistemi critici: > 60% qualifica

Traito da: Ian Sommerville, *Software Engineering*, 6ª ed.


Dipartimento di Informatica, Università di Pisa 23/25

Il ciclo di vita del *software*

## Riepilogo

- ❑ **Il ciclo di vita del software**
- ❑ **Il modello a cascata**
- ❑ **I modelli iterativi**
- ❑ **Il modello a spirale**
- ❑ **Cicli di vita espressi mediante ISO/IEC 12207**

Dipartimento di Informatica, Università di Pisa 24/25

|   |   |
|---|---|
|  | <p>Il ciclo di vita del <i>software</i></p> <h3>Riferimenti</h3> <ul style="list-style-type: none"><li>❑ B.W. Bohem, "A spiral model of software development and enhancement", IEEE Software, maggio 1998</li><li>❑ Center for Software Engineering,<br/><a href="http://sunset.usc.edu/research/spiral_model">http://sunset.usc.edu/research/spiral_model</a></li><li>❑ W.W. Royce, "Managing the development of large software systems: concepts and techniques",<br/>Atti della conferenza "Wescon '70", agosto 1970</li><li>❑ ISO/IEC TR 15271:1998, Information Technology – Guide for ISO/IEC 12207</li></ul> |
| <p>Dipartimento di Informatica, Università di Pisa</p>                            | <p>25/25</p>  |