




Il ciclo di vita del SW



Ingegneria del Software
V. Ambriola, G.A. Cignoni,
C. Montangero, L. Semini
Aggiornamenti : T. Vardanega (UniPD)

Dipartimento di Informatica, Università di Pisa

1/34




Il ciclo di vita del SW
Il concetto di ciclo di vita – 1

- ❑ **Concezione → sviluppo → utilizzo → ritiro**
 - Gli stati (principali) assunti da un prodotto SW durante la sua esistenza
 - Noi ci concentriamo sul segmento [concezione → sviluppo]
- ❑ **La transizione tra stati avviene tramite l'esecuzione di attività di processi di ciclo di vita**
 - Le attività di processo prevedono specifiche responsabilità, che diventano ruoli in un *team* di progetto
- ❑ **Per organizzare le attività dei processi attivati nel progetto**
 - Identifichiamo le dipendenze tra i loro ingressi e le loro uscite
 - Fissiamo il loro ordinamento nel tempo e i criteri di attivazione (pre-condizioni) e di completamento (post-condizioni)

Dipartimento di Informatica, Università di Pisa

2/34



Il ciclo di vita del SW
Il concetto di ciclo di vita – 2

- ❑ **Lo stazionamento in uno stato di ciclo di vita o in una transizione tra stati viene detta fase**
 - “Fase” designa un segmento temporale contiguo, con caratteristiche coerenti
- ❑ **Aderire a un modello di ciclo di vita comporta vincoli sulla pianificazione e gestione del corrispondente progetto**
 - La scelta del modello influenza la selezione del *way of working* e dei suoi strumenti di supporto

Dipartimento di Informatica, Università di Pisa

3/34




Il ciclo di vita del SW
Modelli di ciclo di vita (/ di sviluppo)

- ❑ **Parliamo di “modelli” al plurale, in quanto esiste una varietà di cicli di vita**
 - Che differiscono per transizioni tra stati e regole di attivazione
- ❑ **Nell'esempio vediamo un ciclo di vita che non contempla ritiro**
- ❑ **Come ci sono modelli di ciclo di vita, così vi sono modelli di sviluppo**



Dipartimento di Informatica, Università di Pisa


4/34



Il ciclo di vita del SW
Glossario

- ❑ **Iterazione**
 - Procedere per raffinamenti o rivisitazione (analogo alla pittura)
- ❑ **Incremento**
 - Procedere per aggiunte successive a un impianto base (analogo alla scrittura)
- ❑ **Prototipo**
 - Per provare e scegliere soluzioni
 - Può essere usa-e-getta (iterativo) oppure rappresentare uno stato di avanzamento incrementale (*baseline*)
- ❑ **Riuso**
 - Occasionale: copia-incolla opportunistico → basso costo, scarso impatto
 - Sistemático (per progetto / prodotto / azienda) → maggior costo, forte impatto

Dipartimento di Informatica, Università di Pisa5/34



Il ciclo di vita del SW
Evoluzione dei modelli di sviluppo

- ❑ Dal “Code-’n-Fix” originario, che è un non-modello
 - Insieme di attività senza organizzazione preordinata
 - Fonte di progetti caotici difficilmente gestibili
- ❑ Ai modelli organizzati post-’68
 - **Cascata** rigide fasi sequenziali
 - **Incrementale** realizzazione in più passi
 - **Evolutivo** ripetute iterazioni che rilasciano diverse varianti di prodotto
 - **A componenti** orientato al riuso
 - **Agile** altamente dinamico, fatto di brevi cicli iterativi e incrementali

Dipartimento di Informatica, Università di Pisa6/34



Il ciclo di vita del SW
Modello sequenziale (a cascata) – 1

- ❑ **Definito nel 1970 da Winston W. Royce**
 - “*Managing the development of large software systems: concepts and techniques*”
 - Centrato sull’idea di processi ripetibili
- ❑ **Successione di fasi rigidamente sequenziali**
 - Non ammette ritorno a fasi precedenti
 - Eventi eccezionali fanno ripartire dall’inizio
- ❑ **Prodotti**
 - Principalmente documenti, fino poi a includere il SW
 - L’emissione e l’approvazione di documenti sono condizione necessaria per l’avvio della fase successiva

Dipartimento di Informatica, Università di Pisa7/34



Il ciclo di vita del SW
Modello sequenziale (a cascata) – 2

- ❑ **Ogni stato (fase) è caratterizzato da pre-condizioni di ingresso e post-condizioni di uscita**
 - Il loro soddisfacimento è dimostrato primariamente tramite prodotti documentali e infine tramite esecuzione del SW
- ❑ **Le fasi sono distinte (canoniche) e non sovrappongono nel tempo**
- ❑ **Modello adatto allo sviluppo di sistemi complessi sul piano organizzativo**
 - Le iterazioni costano troppo per essere un buon mezzo di mitigazione dei rischi tramite approssimazioni successive

Dipartimento di Informatica, Università di Pisa8/34

Il ciclo di vita del SW

Modello sequenziale (a cascata) – 3

- ❑ Ogni fase viene definita in termini di
 - Attività previste e prodotti attesi in ingresso e in uscita
 - Contenuti e struttura dei documenti
 - Responsabilità e ruoli coinvolti
 - Scadenze di consegna dei prodotti
- ❑ Entrare, uscire, stazionare in una fase comporta lo svolgimento di determinate azioni
 - Realizzate come attività erogate da specifici processi

Dipartimento di Informatica, Università di Pisa

9/34

Il ciclo di vita del SW

Schema secondo ISO 12207:1995

Dipartimento di Informatica, Università di Pisa

10/34

Il ciclo di vita del SW

Critica del modello sequenziale

- ❑ Difetto principale: eccessiva rigidità
 - Stretta sequenzialità tra fasi (nessun parallelismo e nessun ritorno)
 - Non ammette modifiche nei requisiti in corso d'opera
 - Esprime una visione burocratica e poco realistica del progetto
- ❑ Correttivo 1: con prototipazione
 - Prototipi di tipo “usa e getta”
 - Per capire meglio i requisiti o le soluzioni
 - Strettamente all'interno di singole fasi
- ❑ Correttivo 2: cascata con ritorni
 - Ogni ciclo di ritorno raggruppa sotto-sequenze di fasi

Dipartimento di Informatica, Università di Pisa

11/34

Il ciclo di vita del SW

Ritorni: iterazione o incremento?

- ❑ Non sempre gli *stakeholder* hanno ben chiaro dall'inizio ogni aspetto del sistema richiesto
 - In tal caso bisogna decidere cosa sia meglio fare per aiutarli
 - Per problemi particolarmente complessi conviene prevedere iterazioni
 - Le iterazioni possono essere distruttive e annullare lavoro precedente
- ❑ Non conviene posticipare troppo l'integrazione di tutte le parti del sistema (*big-bang integration*)
 - Meglio l'integrazione continua di piccole parti, che è un tipico procedimento incrementale
- ❑ Iterazione e incremento coincidono quando la sostituzione raffina ma non ha impatto sul resto

Dipartimento di Informatica, Università di Pisa


12/34

 Il ciclo di vita del SW


Vantaggi dei modelli incrementali

- ❑ Possono produrre valore a ogni incremento
 - Un insieme di funzionalità diventa presto disponibile
 - I primi incrementi possono essere frutto di prototipazione, aiutando a fissare meglio i requisiti per gli incrementi successivi
- ❑ Ogni incremento riduce il rischio di fallimento
 - Senza però azzerarlo a causa dei costi aggiuntivi derivanti dalla caduta nell'iterazione
- ❑ Le funzionalità fondamentali vanno sviluppate nei primi incrementi
 - Così che esse siano più frequentemente verificate così diventando via via più stabili


Dipartimento di Informatica, Università di Pisa 13/34

 Il ciclo di vita del SW

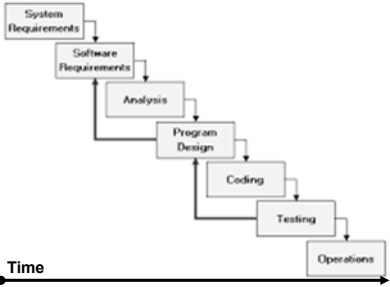
Vantaggi dei modelli iterativi

- ❑ Sono applicabili a qualunque modello di ciclo di vita
 - Con opportuni vincoli
- ❑ Consentono maggior capacità di adattamento
 - Evoluzione di problemi, requisiti utente, soluzioni e tecnologie
- ❑ Ma comportano il rischio di non convergenz: 
- ❑ Soluzione generale
 - Decomporre la realizzazione del sistema
 - Identificare e trattare **prima** le parti più critiche
 - Quelle più complesse oppure quelle i cui requisiti vanno maggiormente chiariti
 - Limitando superiormente il numero di iterazioni

Dipartimento di Informatica, Università di Pisa 14/34

 Il ciclo di vita del SW

Rischi dei modelli iterativi



Ogni iterazione comporta un ritorno all'indietro nella direzione opposta all'avanzamento del tempo

Dipartimento di Informatica, Università di Pisa 15/34

 Il ciclo di vita del SW

Modello incrementale – 1

- ❑ Prevede rilasci multipli e successivi
 - Ciascuno realizza un incremento di funzionalità
- ❑ I requisiti sono classificati e trattati in base alla loro importanza strategica
 - I primi incrementi puntano a soddisfare i requisiti più importanti sul piano strategico
 - Così i requisiti importanti diventano presto chiari e stabili, quindi più facilmente soddisficibili
 - Quelli meno importanti hanno invece più tempo per stabilizzarsi e armonizzarsi con lo stato del sistema

Dipartimento di Informatica, Università di Pisa 16/34

Il ciclo di vita del SW

Schema generale

I cicli di incremento sono parte dello sviluppo

La validazione è anch'essa incrementale se il rilascio è pubblico

Tratto da: Ian Sommerville, *Software Engineering*, 8th ed.

Dipartimento di Informatica, Università di Pisa

17/34

Il ciclo di vita del SW

Modello incrementale – 2

- ❑ **Analisi e progettazione architetturale non vengono ripetute**
 - I requisiti principali sono identificati e fissati completamente
 - L'architettura del sistema è identificata dall'inizio e fissata definitivamente
 - Questo vincolo è essenziale per poter pianificare preventivamente i cicli di incremento
- ❑ **La realizzazione è incrementale**
 - Attività di progettazione di dettaglio, codifica e prove
 - Prima i requisiti essenziali poi quelli desiderabili
 - Integrazione, validazione sui requisiti, eventuale rilascio

Di questo parleremo ampiamente più avanti

Dipartimento di Informatica, Università di Pisa

18/34

Il ciclo di vita del SW

Schema secondo ISO 12207:1995

Dipartimento di Informatica, Università di Pisa

19/34

Il ciclo di vita del SW

Modello evolutivo – 1

- ❑ **Aiuta a rispondere a bisogni non inizialmente preventivabili**
- ❑ **Può richiedere il rilascio e il mantenimento di più versioni esterne attive in parallelo**
- ❑ **Comporta il riattraversamento di più stati di ciclo di vita**

Dipartimento di Informatica, Università di Pisa

20/34

Il ciclo di vita del SW

Modello evolutivo – 2

- ❑ **Analisi preliminare**
 - Identificare i requisiti di massima
 - Definire l'architettura di massima
 - Pianificare i passi di analisi e realizzazione evolutiva
- ❑ **Analisi e realizzazione di una singola evoluzione**
 - Per raffinamento ed estensione dell'analisi iniziale
 - Per progettazione, codifica, prove, integrazione
- ❑ **Rilascio di versioni sempre più complete**
 - Fintanto che necessario

Dipartimento di Informatica, Università di Pisa

21/34

Il ciclo di vita del SW

Schema generale

Tratto da: Ian Sommerville, *Software Engineering*, 8th ed.

Dipartimento di Informatica, Università di Pisa

22/34

Il ciclo di vita del SW

Schema secondo ISO 12207:1995

Dipartimento di Informatica, Università di Pisa

23/34


Il ciclo di vita del SW

Modello a componenti

- ❑ **Molto di quello che ci serve fare è già stato fatto e molto di quello che faremo ci potrà servire ancora**
 - L'analisi dei requisiti viene adattata alle possibilità di riuso
- ❑ **Massima attenzione al riuso sistematico di componenti preesistenti proprie oppure di terzi ("off-the-shelf")**

Dipartimento di Informatica, Università di Pisa

24/34




Il ciclo di vita del SW

Metodi agili – 1

- ❑ **Nascono alla fine degli '90 come reazione alla eccessiva rigidità dei modelli allora prevalenti**
 - <http://agilemanifesto.org/>
- ❑ **Si basano su quattro principi fondanti**
 - **Individuals and interactions over processes and tools**
 - L'eccessiva rigidità ostacola l'emergere del valore
 - **Working software over comprehensive documentation**
 - La documentazione non sempre corrisponde a SW funzionante
 - **Customer collaboration over contract negotiation**
 - L'interazione con gli stakeholder va incentivata e non ingessata
 - **Responding to change over following a plan**
 - La capacità di adattamento al cambiare delle situazioni è importante

Dipartimento di Informatica, Università di Pisa

25/34




Il ciclo di vita del SW

Osservazioni

- ❑ **SW senza documentazione è un costo, non un valore**
 - Commentare il codice non basta → serve motivare e spiegare le scelte realizzative
- ❑ **Senza un piano non si possono valutare rischi e avanzamenti**
 - La sola misurazione di consuntivo non può bastare
- ❑ **Cambiare si può, ma con consapevolezza del rapporto costo/benefici**

Dipartimento di Informatica, Università di Pisa

26/34




Il ciclo di vita del SW

Metodi agili – 2

- ❑ **L'idea base è il concetto di “user story”**
 - Una funzionalità significativa che l'utente vuole realizzare con il SW richiesto
- ❑ **Ogni “user story” è definita da**
 - Un documento di descrizione del problema individuato
 - La minuta delle conversazioni con gli *stakeholder* effettuate per discutere il problema e comprenderlo insieme
 - La strategia da usare per confermare che il SW realizzato soddisfi gli obiettivi del problema

Dipartimento di Informatica, Università di Pisa

27/34



Il ciclo di vita del SW

Metodi agili – 3

- ❑ **Migliori assunti base**
 - Suddividere il lavoro in piccoli incrementi a valore aggiunto che possono anche essere sviluppati indipendentemente
 - Sviluppare gli incrementi in una sequenza continua dall'analisi all'integrazione
- ❑ **Obiettivi strategici**
 - Poter costantemente dimostrare al cliente quanto è stato fatto
 - Verificare l'avanzamento tramite progresso reale
 - Dare agli sviluppatori la soddisfazione del risultato
 - Dimostrare che l'intero prodotto SW è ben integrato e verificato
- ❑ **Esempi**
 - Scrum (organizzazione dietro caos apparente), Kanban (*just-in-time*), Scrumban

Dipartimento di Informatica, Università di Pisa

28/34

Il ciclo di vita del SW

Scrum – 1

- **Product Backlog**
Requisiti e funzionalità del prodotto
- **Sprint Backlog**
Insieme di storie del prossimo sprint

- **Sprint**
Fase operativa di sviluppo
Durata media 2 - 4 settimane
Prodotto potenzialmente vendibile

Dipartimento di Informatica, Università di Pisa

29/34

Il ciclo di vita del SW

Scrum – 2

- **Sprint Planning**
Pianificazione dello sprint
- **Sprint Review**
Controllo prodotti dello sprint

- **Daily Scrum**
Controllo giornaliero avanzamento
- **Sprint Retrospective**
Controllo qualità sullo sprint

Dipartimento di Informatica, Università di Pisa

30/34

Il ciclo di vita del SW

Il ciclo di vita secondo SEMAT /1

Stakeholders

Recognized
Represented
Involved
In Agreement
(Satisfied for Deployment)
Satisfied in Use

Requirements

Conceived
Bounded
Coherent
Acceptable
Addressed
Fulfilled

www.ivarjacobson.com/semat

Sheet 1 of 2

Dipartimento di Informatica, Università di Pisa

31/34

Il ciclo di vita del SW

Il ciclo di vita secondo SEMAT /2

Software System

Architecture Selected
Demonstrable
Useable
Ready
Operational
Retired

Team


Seeded
Formed
Collaborating
Performing
Adjourned

www.ivarjacobson.com/semat

Sheet 2 of 2

Dipartimento di Informatica, Università di Pisa

32/34




Il ciclo di vita del SW

Ripartizione dei costi nei modelli

- ❑ Applicazioni “normali”
 - ~ 60% → realizzazione
 - ~ 40% → 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

Tratto da: Ian Sommerville, *Software Engineering*, 8th ed.

Dipartimento di Informatica, Università di Pisa33/34



Il ciclo di vita del SW

Riferimenti

- ❑ W.W. Royce, “Managing the development of large software systems: concepts and techniques”, Atti della conferenza “Wescon '70”, agosto 1970
- ❑ B.W. Boehm, “A spiral model of software development and enhancement”, IEEE Software, maggio 1998
- ❑ Center for Software Engineering, http://sunset.usc.edu/research/spiral_model
- ❑ ISO/IEC TR 15271:1998, Information Technology – Guide for ISO/IEC 12207
- ❑ Scrum: http://www.scrumalliance.org/learn_about_scrum

Dipartimento di Informatica, Università di Pisa34/34