

Copyright © 2004 Moreno Marzolla

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 2.5 Italy License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.5/it/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Processi di produzione del Software

- Insieme coerente di attività per la specifica, il progetto, l'implementazione, la verifica di sistemi software
- Un *modello* di processo è una rappresentazione astratta di un processo.
 - Descrive un processo da una particolare prospettiva
- Modelli di processo generici:
 - A cascata (*waterfall*)
 - Sviluppo Evolutivo
 - Sviluppo Formale
 - Meccanismi basati sul riuso

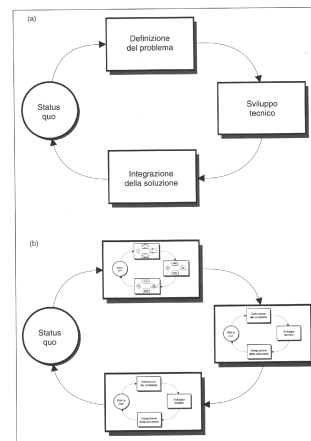
Moreno Marzolla

Ingegneria del Software

2

Il ciclo di risoluzione di un problema / 1

- Status quo
 - Stato delle cose attuale
- Definizione del problema
 - Individua lo specifico problema da risolvere
- Sviluppo tecnico
 - Risoluzione del problema mediante una opportuna tecnologia
- Integrazione
 - Consegna dei risultati al committente



Moreno Marzolla

Ingegneria del Software

3

Il ciclo di risoluzione di un problema / 2

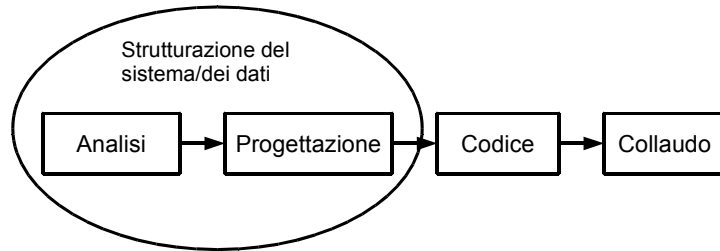
- Il ciclo si può applicare a diversi livelli di dettaglio
 - A livello più alto, all'intero sistema
 - Ma anche ai sottosistemi...
 - ...fino alle singole righe di codice
- Di conseguenza, il ciclo assume un aspetto *frattale (autosimile)*

Moreno Marzolla

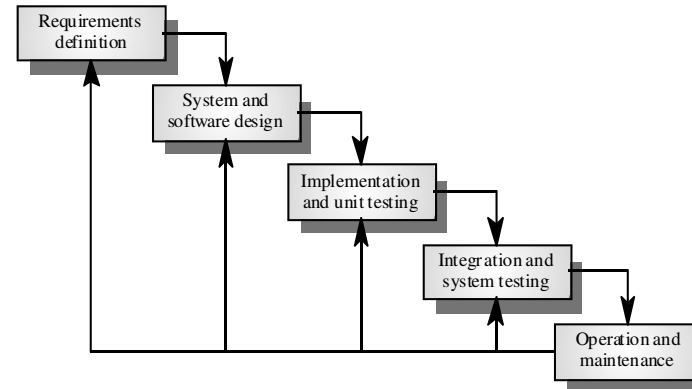
Ingegneria del Software

4

Il Modello a Cascata o Modello Sequenziale Lineare



Il modello a cascata visione più dettagliata



Fasi del modello a cascata

- Analisi dei requisiti
- Disegno del sistema e/o del software
- Implementazione e test dei sottosistemi
- Integrazione e test del sistema
- Operazione e mantenimento del sistema

Visibilità del processo a cascata

Attività	Documentazione prodotta
Analisi dei Requisiti	Studio di fattibilità, Schema dei requisiti
Definizione dei Requisiti	Documento dei Requisiti
Specifica del sistema	Specifica Funzionale, Schema manuale utente
Disegno dell'Architettura	Specifica Architetture, Piano dei test sull'Architettura
Disegno dell'Interfaccia	Specifica dell'Interfaccia, Piano dei test sull'Interfaccia
Disegno dettagliato	Specifica di Disegno, Definizione dei test delle singole unità
Codifica	Codice del programma
Test delle unità	Rapporto del test
Test Integrazione componenti	Rapporto del test, Manuale utente finale
Test del Sistema	Rapporto del test
Test di accettazione	Sistema finale e documentazione

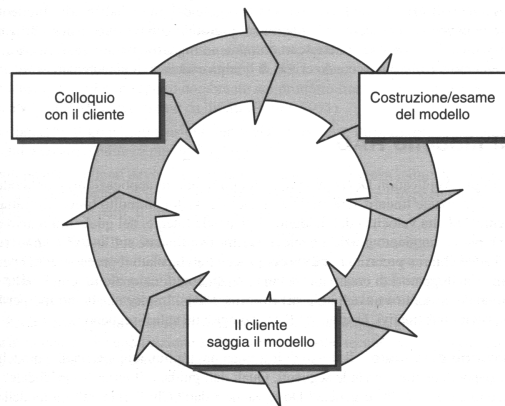
Problemi del modello a cascata

- Difficile operare cambiamenti una volta che il processo è in corso
 - Pertanto è difficile soddisfare cambiamenti nei requisiti da parte del committente.
- Il partizionamento in fasi può apparire *arbitrario* o *artificioso*
 - In ogni caso il modello a cascata è basato sul ben consolidato modello ingegneristico per la risoluzione dei problemi
- Questo modello è adeguato quando i requisiti sono ben compresi (e non soggetti a modifiche)

La Prototipazione / 1

- Spesso i requisiti non sono abbastanza chiari
 - Il cliente stabilisce gli scopi generali del sistema software, ma non ne chiarisce subito i requisiti in modo dettagliato
 - I programmatori sono incerti sul significato dei requisiti, o su come strutturare l'interfaccia o gli algoritmi
- In tali casi viene in aiuto una metodologia di sviluppo basata su *prototipi* (paradigma prototipale)

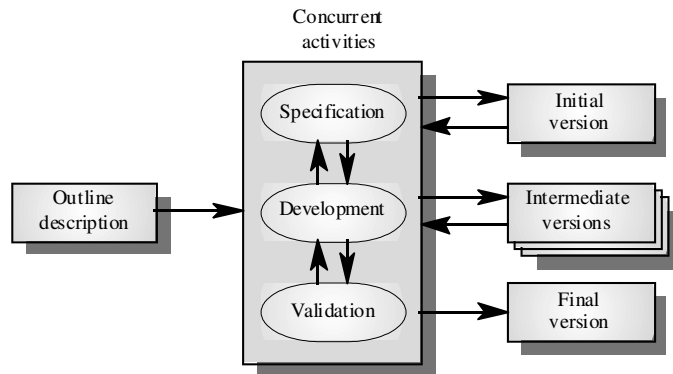
La Prototipazione / 2



Sviluppo Evolutivo

- *Exploratory Programming*
 - L'idea è lavorare con il committente evolvendo un prodotto finale a partire dai requisiti.
 - Nuove caratteristiche vengono aggiunte man mano che sono proposte dal cliente.
 - Il sistema viene costruito a partire dai requisiti che sono **ben compresi**
- *Throw-Away Prototyping*
 - Obiettivo è capire meglio i requisiti del cliente
 - Il punto di partenza sono i requisiti che **non sono ben compresi**

Schema dello sviluppo evolutivo



Sviluppo Evolutivo / 1

- Problemi
 - Mancanza di visibilità (quando produrre i documenti?)
 - I sistemi prodotti sono spesso poco strutturati
 - Competenze particolari sono spesso richieste (es. in linguaggi per la prototipazione rapida)
- Occorre resistere alla tentazione di trasformare il prototipo in un prodotto da porre in produzione
 - Al cliente il prototipo potrebbe *apparire* funzionante, ma in realtà potrebbe avere tali e tanti problemi da renderlo di fatto inutilizzabile in condizioni reali.
 - Quindi il sistema va rifatto

Sviluppo Evolutivo / 2

- Quando applicarlo
 - Per sistemi di piccole o medie dimensioni
 - Per parti specifiche di sistemi (es. interfacce utente)
 - Per sistemi con vita attesa corta
 - Per sviluppare prototipi

Che fare del prototipo alla fine?

Nella maggior parte dei progetti, il sistema realizzato per primo è a malapena utilizzabile. Può risultare troppo lento, troppo grande, di uso complicato o tutte e tre le cose. Non c'è altro da fare che ricominciare da capo, facendo tesoro dell'esperienza e realizzare una nuova versione, nella quale i problemi siano risolti... Quando si applica una nuova concezione o una nuova tecnologia, è inevitabile costruire un sistema pilota destinato a essere cestinato, perché anche con la migliore pianificazione non si può prevedere tutto e centrare il bersaglio al primo colpo. La questione non è dunque se costruire un sistema pilota e poi gettarlo. Ciò è inevitabile. La questione è se pianificare in anticipo di costruire un oggetto destinato a essere gettato, oppure promettere al cliente di consegnargli quell'oggetto

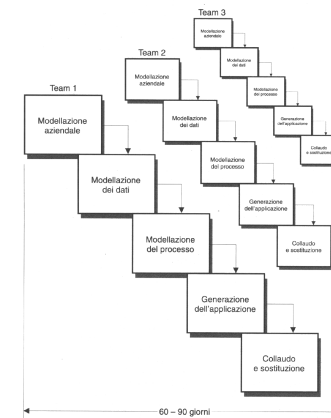
Frederick P. Brooks, Jr.
"The Mythical Man-Month"

Il modello RAD / 1 (Rapid Application Development)

- E' un modello sequenziale lineare che punta ad un ciclo di sviluppo molto breve
- L'obbiettivo di uno sviluppo rapido è raggiunto mediante il riuso di componenti
- Quando applicarlo
 - Quando i requisiti sono chiari
 - Quando il processo di sviluppo è ben vincolato
- Nei casi in cui si può applicare, il modello RAD può portare allo sviluppo del software in tempi brevi rispetto al modello a cascata classico

Il modello RAD / 2

- E' adatto alle situazioni in cui il sistema da costruire può essere partizionato facilmente fin dall'inizio
- Ogni parte deve poter essere sviluppata indipendentemente dalle altre, e in tempi brevi (<3 mesi)



Le fasi del modello RAD / 1

- Modellazione aziendale
 - Rispondere alle seguenti domande: quali dati guidano il processo aziendale? Quali dati vengono generati? Chi li genera? Dove vengono utilizzati?
- Modellazione dei dati
 - I dati di cui al punto precedente vengono raffinati in una serie di "oggetti". Di ciascuno si indica la struttura interna e la relazione con altri oggetti
- Modellazione del processo
 - Si definiscono le operazioni compiute sui dati (inserimenti, cancellazione, modifica), in base alle necessità del sistema da implementare

Le fasi del modello RAD / 2

- Generazione dell'applicazione
 - L'applicazione viene generata usando le cosiddette "tecniche di quarta generazione" (linguaggi di alto livello, riuso di componenti già esistenti o creazione di nuove componenti riutilizzabili)
- Collaudo e ricambio
 - Molte componenti del sistema sono state riutilizzate da altri progetti, quindi sono già testate. Le nuove componenti, invece, vanno collaudate e l'intero sistema va verificato (ricordate le *proprietà emergenti*...)

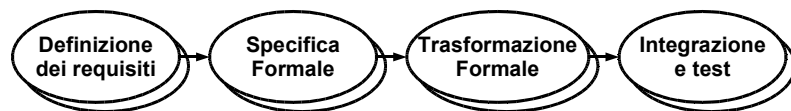
I limiti del modello RAD

- Per progetti di grandi dimensioni (purché partizionabili) è necessario disporre di risorse umane sufficienti
- Sviluppatori e clienti devono impegnarsi a completare il sistema in tempi rapidi
- RAD è inadatto quando:
 - ...Il sistema non è partizionabile
 - ...Le prestazioni sono un aspetto cruciale e dipendono da interfacce appositamente definite
 - ...Si sviluppano sistemi con tecnologie innovative, con alto rischio di incontrare problemi strada facendo

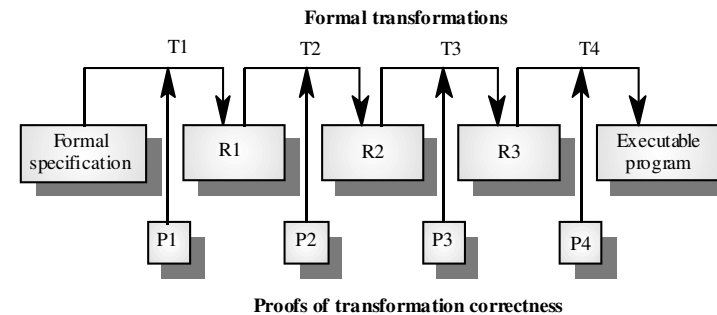
Sviluppo Formale

- Si basa sulla trasformazione di specifiche matematiche di un sistema in differenti rappresentazioni
- Le trasformazioni devono preservare la correttezza
 - In tal modo è banale verificare che il prodotto finale soddisfa la specifica

Fasi dello Sviluppo Formale



Trasformazioni formali



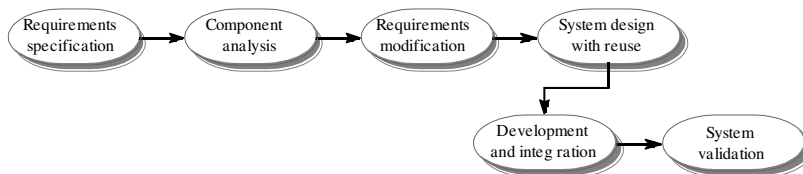
Sviluppo Formale

- Problemi
 - Richiede conoscenze specializzate e addestramento per essere applicato
 - Certe parti del sistema (es. interfaccia utente) sono difficili da specificare formalmente
- Quando applicarlo
 - Sistemi critici, in cui sicurezza o affidabilità sono essenziali e devono essere *provabilmente* raggiunti

Sviluppo basato sul riuso di componenti

- Basato sul sistematico riuso di componenti *off-the-shelf*, integrate opportunamente
- Fasi del processo
 - Analisi delle componenti
 - Adattamento dei requisiti
 - Disegno del sistema
 - Integrazione

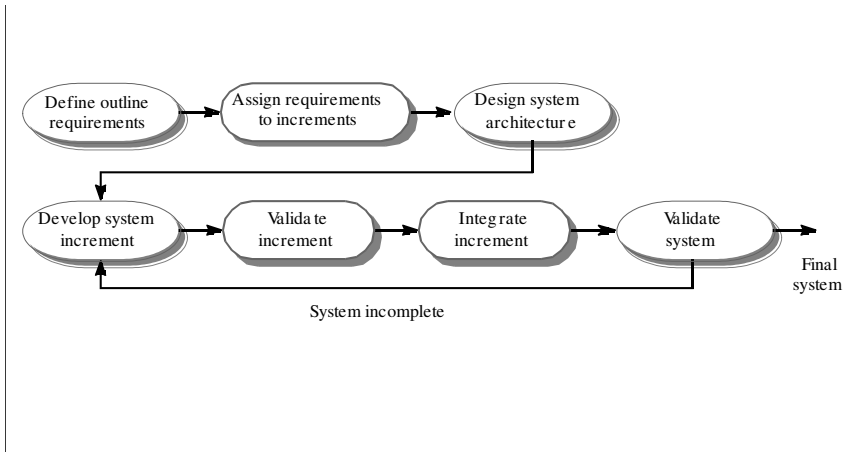
Sviluppo basato sul riuso



Sviluppo incrementale

- Combina idee dello sviluppo basato su prototipi con quello lineare sequenziale
- Il sistema non viene fornito completo da subito
 - Viene sviluppato un prototipo che soddisfa parte dei requisiti
 - Ad ogni iterazione si aggiungono nuove funzionalità
 - I requisiti del cliente vengono realizzati a partire da quelli con priorità più alta
 - I requisiti vengono “congelati” durante ogni iterazione

Lo Sviluppo Incrementale



Vantaggi dello Sviluppo Incrementale

- I requisiti a più alta priorità tendono ad ricevere maggiori test
 - Perché sono implementati per primi e testati ad ogni iterazione
- Il cliente riceve presto un prototipo funzionante, e può fornire feedback
 - I feedback vengono usati per l'iterazione successiva
- Minor rischio di fallimento dell'intero processo

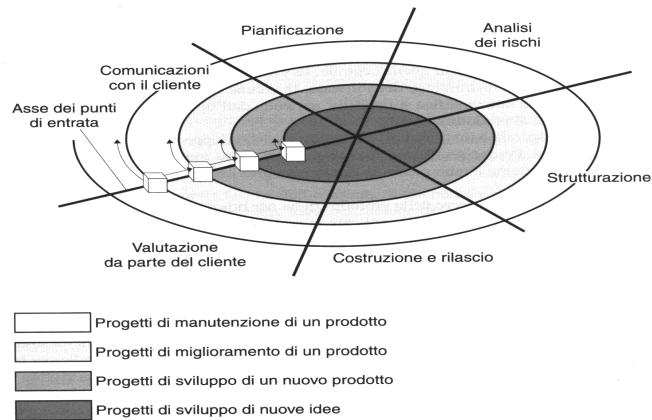
Extreme Programming (XP)

- Approccio recentissimo basato sulla consegna pressoché continua di prototipi con piccoli incrementi di funzionalità
- Si basa sul miglioramento costante del codice e sulla partecipazione degli utenti per avere feedback

Modello a Spirale

- Il processo di sviluppo è rappresentato come una spirale, piuttosto che come una sequenza
- Ogni ciclo nella spirale è una fase del processo
- Al termine di ogni "giro" il risultato può essere un *progetto*, un *prototipo*, un *sistema funzionante* o un *prodotto software completo*
- Non ci sono fasi predefinite. Il management del progetto deve decidere come strutturarli in fasi

Schema del modello a Spirale



Sviluppo a spirale

- La spirale è divisa a “spicchi”
 - **Comunicazione con il Cliente.** Specificare gli obiettivi e i vincoli di quella fase, identificare i rischi ed eventualmente proporre delle strategie alternative
 - **Pianificazione.** Attività rivolte a definire scadenze e risorse
 - **Analisi dei rischi.** Attività rivolte a stimare i rischi tecnici e di gestione
 - **Strutturazione.** Attività rivolte a costruire una o più rappresentazioni (strutture alternative) del sistema
 - **Costruzione e rilascio.** Attività di sviluppo, effettuata secondo un modello generico (cascata, evolutivo...)
 - **Valutazione da parte del cliente.** Attività rivolte a ricevere le reazioni del cliente su quanto fin qui realizzato (nei primi giri della spirale si creano “idee” o “proposte”, nei successivi anche “prodotti” o “prototipi”)

Analisi dei Rischi

- Il modello a spirale, a differenza degli altri, tiene conto dei *rischi*
- **Rischio** è evento imprevisto che può causare problemi o difficoltà
 - Esempio: Stiamo usando un compilatore per un nuovo linguaggio. C'è il rischio che il compilatore sia bacato.
- I rischi sono conseguenza di informazioni insufficienti
- Si risolvono acquisendo maggiori informazioni per ridurre l'incertezza

Modello a Spirale

- **Vantaggi**
 - Valutazione esplicita dei rischi
 - Adatto allo sviluppo di sistemi complessi di grandi dimensioni
 - Interazione con il cliente ad ogni giro della spirale
- **Svantaggi**
 - La stima dei rischi richiede competenze specifiche
 - Se i rischi non vengono valutati correttamente, potranno sorgere problemi nelle fasi successive
 - Il modello è relativamente nuovo e poco sperimentato

Confronto tra le visibilità dei modelli dei processi

- Modello a Cascata
 - Ottima visibilità
- Modello Evolutivo
 - Visibilità scarsa. E' antieconomico produrre documentazione a ritmo continuo durante le iterazioni
- Modello a Trasformazioni Formali
 - Buona visibilità (suddiviso in fasi)
- Modello basato sul riuso
 - Moderata visibilità. Può essere "eccessivo" produrre documentazione su *come* riutilizzare le componenti
- Modello a Spirale
 - Buona visibilità. Ogni segmento e ogni giro della spirale produce documenti