



## 3. Ciclo di Vita e Processi di Sviluppo come posso procedere nello sviluppo?

Andrea Polini

Ingegneria del Software  
Corso di Laurea in Informatica

# Sommario

- 1 Il ciclo di vita del software
- 2 Processi di Sviluppo Software
- 3 Computer-Aided Software Engineering

# Sommario

- 1 Il ciclo di vita del software
- 2 Processi di Sviluppo Software
- 3 Computer-Aided Software Engineering

# Process Activities

Tipicamente i differenti processi software si distinguono in base alla organizzazione delle differenti fasi dello sviluppo:

- Software Specification
- Software Design and Implementation
- Software Validation
- Software Evolution

# Software Specification

- Capire cosa il sistema deve fare
- Quali sono i vincoli a cui il sistema deve sottostare
- Differenti tipi di specifica
  - Rivolta al customer
  - Rivolta allo sviluppatore

# Software Specification

...continua

Ingegneria dei requisiti prevede quattro fasi principali:

- Studio di fattibilità
- Elicitazione dei requisiti ed analisi
- Specifica dei Requisiti
- Validazione dei Requisiti

# Software Design and Implementation

Derivare una descrizione:

- del software che deve essere sviluppato e della sua architettura
- dei dati che devono essere scambiati
- componenti facenti parte del sistema
- interfacce tra i vari elementi del sistema
- algoritmi utilizzati

# Software Design and Implementation

Differenti processi postpongono le varie fasi verso l'implementazione o verso il design

## Approcci al design

- Metodologie agili tendono a ridurre i modelli sviluppati nelle fasi
- Metodi strutturati tendono a sviluppare molti modelli usando notazioni grafiche(i.e. UML)

# Software Design and Implementation

...continua

Tipici modelli in approcci strutturati:

- Modello ad oggetti
- Modello di sequenze
- Modello a transizione degli stati
- Modello strutturale
- Modello del flusso dei dati

Debugging è fa parte delle attività di design ed implementation.

Testing vs. Debugging?

# Software Validation

Obiettivo è verificare che il sistema soddisfa i requisiti Tecniche utilizzabili:

- Ispezione del codice
- Testing

Tipicamente testing strutturato su più fasi:

- Testing di componente o di unità
- Testing di integrazione
- Testing di sistema
- Testing di Accettazione (alpha testing)
- Beta testing

# Software Evolution

Riguarda le attività che sono messe in atto sul software già rilasciato.

Tipi di evoluzione:

- correttivo
- adattivo
- perfettivo

# Sommario

- 1 Il ciclo di vita del software
- 2 Processi di Sviluppo Software**
- 3 Computer-Aided Software Engineering

# Modelli di Processo

Nessuna soluzione generale!!

Principali modelli di processo discussi (Attenzione! sono solo categorie):

- waterfall
- Sviluppo Evolutivo
- Component-based Software Engineering
- Sviluppo in camera sterile (cleanroom development, B method, Model Driven Development)

Nella realtà spesso processi utilizzati sono una commistione dei diversi modelli

# Processo a Cascata

- Le attività costituenti il processo sono eseguite sequenzialmente.
- terminata un'attività questa non viene ulteriormente riconsiderata nelle fasi successive. Sovrapposizione dei vari team per passaggio di informazioni.

## Conseguenze principali:

- I requisiti vengono fissati ad un certo punto e mai più modificati
- Prematura decisione sui requisiti rende il processo “sordo” alle successive richieste di adattamento dei requisiti da parte del cliente.
- Processo fortemente documentato (document driven).

Visibilità alta o bassa? Timeliness? Produttività?

Quando applicarlo?

# Processo Evolutivo

Il prodotto viene sviluppato tramite incrementi successivi.

- Sviluppo Esplorativo
- Throaway Prototyping

Conseguenze principali:

- Il sistema tende ad essere poco strutturato (a causa dei continui cambiamenti)
- Induce probabili difficoltà in fasi di mantenimento.

Visibilità alta o bassa? Timeliness? Produttività?

Quando applicarlo?

“Do it twice” - *Baker*

# Component based software engineering

generalità

Aumento nella complessità dei sistemi implica forte spinta al riuso (interno e non - COTS)

Obiettivo di CBSE: implementare un sistema come integrazione di componenti preconfezionati.

Un po' di storia ...

- Conferenza di Garmisch 1968
- primi modelli a componenti OLE-X (anni ottanta)
- oggi ...
  - Modelli di componenti Desktop (COM, Bonomo ...)
  - Modelli di componenti distribuiti (CCM, EJB, ...)

# Component based Software Engineering

## Modelli di componenti

Definiscono una piattaforma di supporto al riuso ed una serie di regole per:

- interfacce da definire
- meccanismi di interazione tra componenti
- impacchettare il componente

# Component based Software Engineering

## Processo di sviluppo

Non si sono ancora evidenziati processi di sviluppo condivisi in questo ambito.

Principali differenze comunque si avranno con introduzione di nuove fasi nello sviluppo e nuove iterazioni.

- Component provisioning
- Requirements adaptation
- Design orientato al riuso
- Sviluppo orientato all'integrazione ed adattamento.

# Component based Software Engineering

## Conseguenze

Visibilità alta o bassa? Timeliness? Produttività?

Quando applicarlo?

Approccio profondamente differente allo sviluppo. Introduzione all'interno di un'organizzazione richiede training e persone con forti doti di astrazione.

Capacità di sviluppare sistemi complessi in minor tempo. Riduce il rischio dello sviluppo. Tendenzialmente il prodotto finale dovrebbe essere "migliore".

# Processi Iterativi

Cercano di fondere i processi evolutivi e a cascata cercando di mantenerne le qualità positive rimuovendo quelle negative.

Basati sul concetto di iterazione:

- Rilascio incrementale
- sviluppo a spirale

# Rilascio incrementale

- Processo procede con tanti mini “waterfall” in sequenza.
- Ad ogni iterazione viene rilasciato un sistema funzionante che potrà essere utilizzato dal cliente.
- Si pianificano iterazioni successive introducendo via via nuove funzionalità.
- Una volta attivata un’iterazione deve essere terminata senza interferenze
- Ogni iterazione dovrebbe tendere ad aumentare il sistema di una porzione gestibile (20.000 LOC?)

# Sviluppo a Spirale

Proposto da Boehm nel 1988 introduce gestione del rischio nella pianificazione delle varie iterazioni.

Rappresentabile su un piano tramite una spirale dove ogni quadrante consiste di attività volte a:

- Specificare gli obiettivi dell'iterazione ed identificazione dei rischi
- Valutare il rischio e definire tecniche per la sua gestione
- Procedere allo sviluppo ed alla validazione
- Pianificazione, si procede a valutare la necessità di un'ulteriore iterazione.

# Processi Iterativi

Visibilità alta o bassa? Timeliness? Produttività?

Quando applicarlo?

# Sommario

- 1 Il ciclo di vita del software
- 2 Processi di Sviluppo Software
- 3 Computer-Aided Software Engineering**

# CASE

Ingegneria del software si occupa anche di sviluppare strumenti di supporto alle varie attività.

Sempre più basati su potenti interfacce grafiche possono essere classificati in:

- Tools: si rivolgono a specifici task
- Workbench: si riferiscono a specifiche attività
- Environments: supportano buona parte del processo.