

Lezione 3

Processi per lo sviluppo rapido del software

Riferimenti bibliografici

- I. Sommerville – Ingegneria del Software – 8a edizione – Cap.17
- R. Pressman- Principi di Ingegneria del Software- 4 edizione- Cap. 3

Argomenti della lezione

- Sviluppo rapido e Processi di sviluppo incrementali ed iterativi.
- Il processo RAD
- I ruoli della prototipazione nel processo di sviluppo software

Motivazioni per lo sviluppo rapido

- A causa dei rapidi cambiamenti dell'ambiente globale in cui operano, le aziende devono rispondere continuamente a nuove opportunità e mercati.
- La rapidità dello sviluppo e consegna è oggi spesso il requisito più critico per i sistemi software.
- Molte aziende sono disposte a transigere sulla qualità, pur di avere una rapida consegna delle funzionalità essenziali.

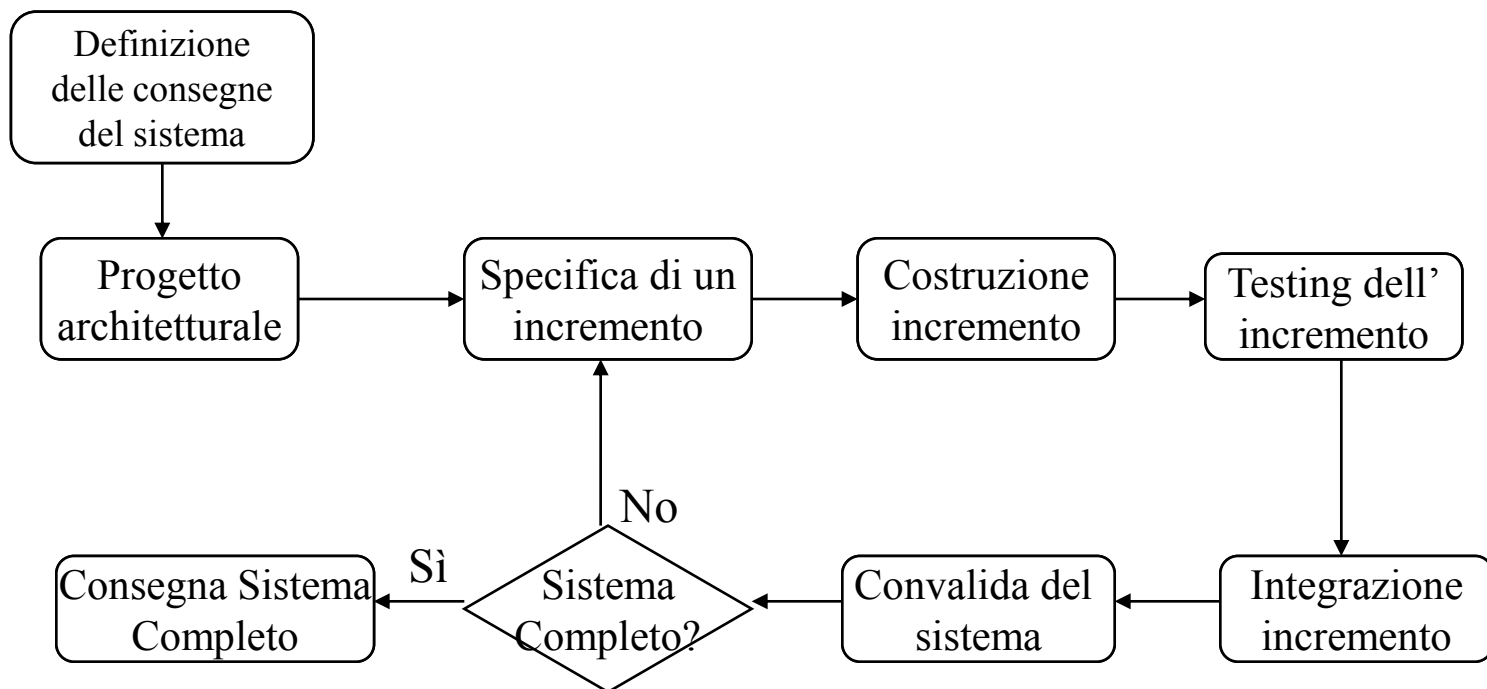
Requisiti

- A causa di un ambiente in continua evoluzione, è spesso impossibile arrivare a requisiti stabili e consistenti da subito: ci si arriverà solo con l'utilizzo del software.
- Un modello di sviluppo a cascata è dunque impraticabile ed un approccio allo sviluppo basato su specifiche e consegna iterative è il solo modo per consegnare il software rapidamente.

Caratteristiche dei processi rapidi

- I processi di specifica, progettazione ed implementazione sono concorrenti. Non esiste una specifica dettagliata e la documentazione di progetto è ridotta al minimo.
- Il sistema è sviluppato **iterativamente** in una serie di **incrementi**. Gli utenti valutano ciascun incremento e quindi propongono modifiche e fanno proposte per i successivi incrementi.
- Le interfacce utente del sistema sono spesso sviluppate in modo interattivo usando ambienti interattivi per la progettazione di UI e la generazione del codice.

Un processo di sviluppo iterativo



Vantaggi dello sviluppo incrementale

- **Consegna rapida dei servizi ai clienti.** Gli incrementi rilasciano dapprima le funzionalità a maggiore priorità per il cliente.
- **Coinvolgimento degli utenti nel sistema.** Gli utenti sono coinvolti nel processo di sviluppo fornendo i propri feedback: di qui, maggiore probabilità di soddisfarne i bisogni, e maggiore volontà degli utenti di fare funzionare il sistema.

Problemi dello sviluppo incrementale

- **Problemi di gestione**
 - Gli avanzamenti del lavoro e gli eventuali problemi possono essere valutati con difficoltà, giacchè non tutta la documentazione di sistema viene prodotta.
- **Problemi contrattuali**
 - É difficile scrivere un contratto senza una specifica; il contratto può essere scritto sulla base del tempo impiegato, ma può essere insoddisfacente sia per i clienti che gli sviluppatori.
- **Problemi di Validazione**
 - Senza una specifica, rispetto a cosa si può verificare e validare il sistema?
- **Problemi di Manutenzione**
 - Le modifiche continue tendono a corrompere la struttura del software, rendendo più difficile le modifiche future.

Applicabilità dello sviluppo incrementale

- Non è usabile per sistemi molto grandi, con più team di sviluppo paralleli e distribuiti, né per sistemi critici, che devono essere analizzati rispetto a tutti i requisiti.
- Per questi sistemi, se c'è incertezza nei requisiti iniziali, si può usare la prototipazione evolutiva (con sviluppo incrementale) con prototipo usa e getta.

Alcuni processi di sviluppo rapido

- Metodi Agili
- RAD [*Martin 91*]
- Metodi basati su riutilizzo di COTS
- Prototipizzazione

Rapid Application Development (RAD)

- Sebbene i metodi agili abbiano ricevuto molta attenzione di recente, già in precedenza si usavano altri metodi rapidi per lo sviluppo di applicazioni.
- Si tratta di approcci pensati per sviluppare applicazioni aziendali di tipo data-intensive, che si basano su un insieme di strumenti per creare, ricercare, visualizzare dati (es. Linguaggi di quarta generazione)
- Il più famoso è l'approccio RAD proposto da James Martin nel 1991

RAD [Martin 91]: obiettivi

- Maggiore velocità:
 - È un modello di processo incrementale che punta ad un ciclo di sviluppo molto breve (60/90 giorni).
- Migliore Qualità
 - La qualità è intesa non come assenza di difetti, ma come capacità dell'applicazione sia di soddisfare i bisogni utente, sia di presentare bassi costi di manutenzione.

Un White paper sull'argomento RAD è scaricabile da:
<http://www.blueink.biz/RapidApplicationDevelopment.aspx>

Elementi essenziali del RAD

- Prototipizzazione
- Sviluppo Iterativo
- Sviluppo Time Boxed (a feature ridotte)
- Team di lavoro
- Management del progetto
- Strumenti RAD

Prototipizzazione

- Lo sviluppo si basa sulla costruzione di un prototipo che serve al cliente come prova dell'applicazione e per raffinare i requisiti.
- Il prototipo viene fatto evolvere iterativamente nel sistema completo.
- Per lo sviluppo del prototipo si fa uso di strumenti CASE che catturano rapidamente i requisiti, li convertono in un modello dei dati, e quindi convertono il modello nel database e generano automaticamente il codice.

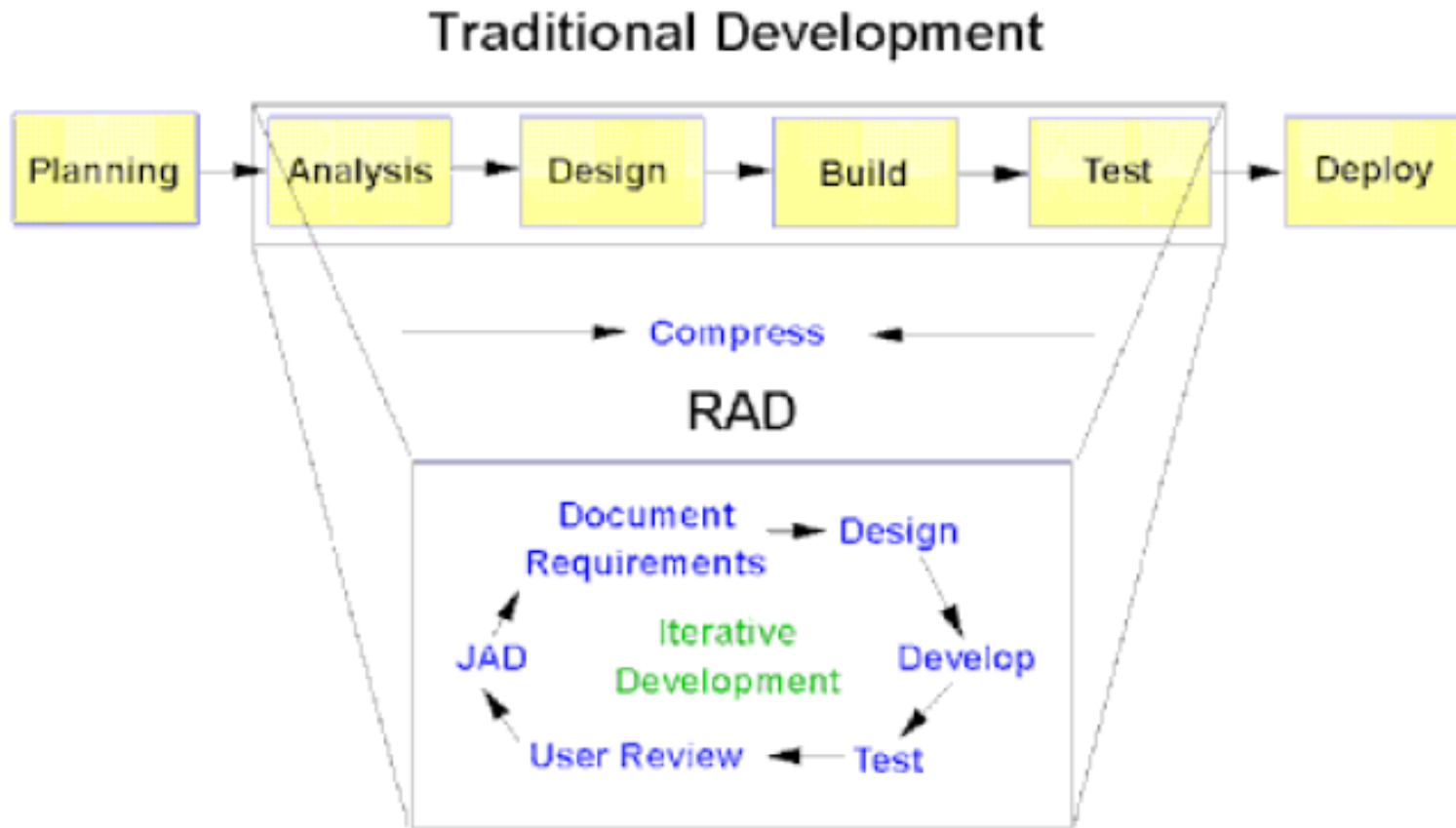
Sviluppo Iterativo e Time Boxing

- Ogni versione viene rivista col cliente per stabilire i requisiti da sviluppare con la prossima versione.
- Per accorciare i tempi di completamento delle versioni ed ottenere rapidamente il feedback del cliente, si selezionano accuratamente le 'core' features da realizzare e si rimandano quelle inessenziali a iterazioni successive (Time boxing).

Team di lavoro e Management

- Si raccomanda l'impiego di **piccoli team di lavoro** con persone motivate e capaci di svolgere ruoli diversi.
 - I team di lavoro possono includere anche il cliente o utenti dell'applicazione (es. per le riunioni di raccolta requisiti, o di progettazione rapida)
 - Gli SWAT (Skilled Workers with Advanced Tools) sono gruppi di esperti in RAD e CASE tools che si occupano dello sviluppo
- Il management deve evitare il rischio di lunghi cicli di lavoro, incomprensioni col cliente, o mancate consegne.
 - Opera attraverso una accurata selezione del personale, motivandole per il successo del progetto, rimuovendo ostacoli politici o burocratici.

Confronto fra Waterfall e RAD



Il Processo RAD

- Basato su 4 fasi:
 1. Pianificazione dei requisiti
 2. User Design (o Functional Design)
 3. Construction
 4. Implementation (o Deployment)

Pianificazione di requisiti e User Design

- Sono raccolti i requisiti utente iniziali e le principali entità di business, definendo lo scope dell'applicazione.
- Sono impiegati strumenti per la gestione dei requisiti, che offrono funzioni di creazione del database a partire dal modello dei dati.
- La fase di User Design si basa su **JAD** (Joint Application Development) workshops (per utenti ed analisti) in cui si modellano i dati ed i processi del sistema (E-R Diagram), si definisce un piano di test e si costruisce eventualmente un prototipo funzionante delle parti più critiche.

Construction

- Il Design Team sviluppa l'applicazione in maniera iterativa, con iterazioni brevi (fra 1 e 3 settimane), convertendo il data model (E-R) nel database.
 - Con appositi CASE le iterazioni sviluppano prototipi funzionanti anche in pochi giorni.
 - Il prototipo viene testato con l'utente e rivisto dagli sviluppatori
 - Al termine, si progetta la prossima iterazione attraverso la definizione dei nuovi requisiti, processi e piani di test.
- Nell'iterazione finale si esegue il test di accettazione

Implementazione

- Il sistema finale viene integrato e rilasciato nell'ambiente operativo
- Gli utenti finali vengono addestrati e svolgono il Test di accettazione finale.
- Si raccolgono i feedback ed eventuali richieste di miglioramenti

Strumenti per gli ambienti RAD

- Linguaggi per la programmazione di Database (es. SQL-che può essere anche generato automaticamente da moduli)
- Generatori di Interfacce (per I/O dati)
- Collegamenti ad applicazioni office
- Generatori di rapporti
 - Esempio: Microsoft Access

Generatori di interfacce

- Molte applicazioni gestionali si basano su moduli strutturati di input e output: sviluppare a mano questi moduli è impraticabile!
- Gli ambienti RAD forniscono in genere aiuti per generare schermate automaticamente, quali:
 - Definizione interattiva di moduli usando tecniche di drag and drop;
 - Collegamento dei vari moduli, specificando l'ordine di presentazione degli stessi;
 - Verifica dei campi dei moduli : i valori consentiti per i campi sono definiti dal programmatore.

Programmazione Visuale

- I linguaggi di scripting tipo Visual Basic sono stati tra i primi a supportare la programmazione visuale che permette di creare un prototipo sviluppando una interfaccia utente a partire da componenti standard (finestre, campi, pulsanti, menu) e associando ad essi delle elaborazioni svolte mediante script.
- Esiste una vasta libreria di componenti software già pronti che supporta questo tipo di sviluppo.
- I componenti possono essere adattati per soddisfare requisiti specifici.

Alcune categorie di strumenti per il RAD

- Cross-Platform RAD tools
 - Plug-in per Eclipse, o NetBeans, ...
- Desktop RAD tools
 - Visual Basic, wxDev-C++ (an extension of Dev-C++)...
- Database RAD Tools
 - IBM Rational, ...
- Embedded Control RAD Tools
- Web Based RAD Tools
 - IBM Rational, Ruby on Rails, Oracle, ...
- Components based on RAD paradigm

http://en.wikipedia.org/wiki/List_of_Rapid_Application_Development_tools

Problemi dello Sviluppo Visuale

- Questo tipo di sviluppo è adatto per applicazioni relativamente semplici, prodotte da piccoli team.
- È invece difficile coordinare lo sviluppo di molti team coinvolti nel processo di sviluppo
- Non c'è una architettura esplicita del sistema.
- Ci sono spesso dipendenze complesse fra le parti del sistema che creano problemi di manutenibilità.

Il Riutilizzo di sistemi COTS

- Una possibile alternativa per lo sviluppo rapido consiste nel configurare e collegare fra loro sistemi applicativi completi (off the shelf).
- Ad esempio, un sistema di gestione dei requisiti potrebbe essere costruito usando:
 - Un database per memorizzare i requisiti;
 - Un word processor per scrivere i requisiti e preparare report;
 - Un foglio elettronico per realizzare la tracciabilità fra i requisiti.