

Processi iterativi

L'inizio delle iterazioni è quasi sempre preceduto da una fase esplorativa di definizione dei requisiti globali del sistema a un livello di astrazione molto elevato: questo passo è necessario per suddividere i requisiti e assegnarli alle iterazioni successive

Ogni iterazione svolge un intero ciclo di sviluppo sw, una sorta di “mini-cascata” (analisi, progettazione, codifica e testing)

Ogni iterazione deve produrre sw finito: anche se il sistema non viene rilasciato all'esterno, la sua qualità dovrebbe essere abbastanza alta da permetterlo

L'iterazione è la chiave per uno sfruttamento efficace dell'OO

Time boxing = tecnica di gestione delle iterazioni, secondo cui ciascuna di esse non deve superare vincoli temporali stabiliti a priori, cioè non si deve mai cambiare la data di conclusione di un'iterazione; all'occorrenza si può rimandare parte delle funzionalità prevista per l'iterazione corrente alla successiva (ciò aiuta anche a capire le vere priorità dei requisiti).

Solitamente si definiscono cicli di uguale durata per tutto lo sviluppo

Unified (Software Development) Process (o Rational Unified Process, RUP) (Jacobson, Booch, Rumbaugh, 1999)

Principi

- Guidato dai casi d'uso, che danno avvio al processo e collegano tutti gli stadi di lavorazione
- Iterativo e incrementale → il sistema completo si ottiene per approssimazioni successive e gli sviluppatori possono lavorare più orientati agli obiettivi, perché le finalità di ogni iterazione sono più dettagliate (e limitate) di quelle finali del progetto
- Centrato sull'architettura: la descrizione dell'architettura inizia parallelamente a quella dei casi d'uso - illustrando, innanzi tutto, la parte che è indipendente dai casi d'uso (per es. la piattaforma di destinazione) - e matura progressivamente nell'ambito dello sviluppo incrementale

RUP: caratteristiche

- struttura per l'Ingegneria del sw a oggetti
- incompatibile col processo a cascata
- dotato di controllo qualità
- attrezzato per la gestione del rischio
- basato su modelli (specificati in UML)

∃ una libreria di istanze di RUP (chiamate casi di sviluppo) già pronte (una è quella proposta dal testo di Zuser et al.)

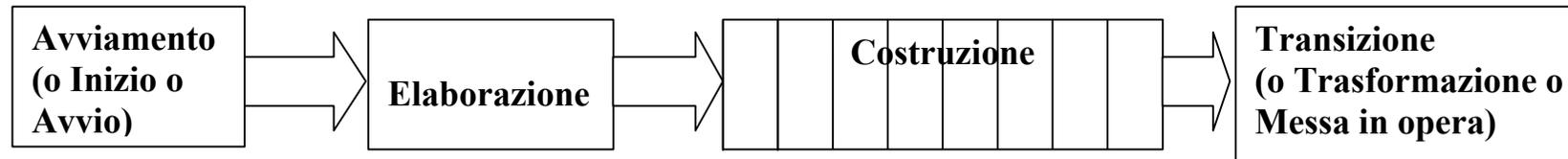
RUP e sviluppo agile

RUP cerca di sfruttare le migliori funzionalità e caratteristiche dei modelli convenzionali di processo, caratterizzandoli però secondo lo sviluppo agile

Agile si riferisce genericamente a processi che

- Hanno una natura fortemente adattiva (cioè la squadra di sviluppo adatta azioni, compiti, sottocompiti e prodotti del processo sulla base delle esigenze contingenti)
- Sono fortemente orientati alle persone (ritengono che il fattore più importante per il successo di un progetto sia la qualità delle persone e la loro collaborazione)
- Tendono a usare iterazioni corte, di durata costante (≤ 1 mese)
- Prevalentemente usano UML per tracciare abbozzi

RUP: fasi



Avviamento: valutazione commerciale (costi/guadagni) del progetto (lo sponsor del progetto si prende l'impegno a sostenere il passaggio alla fase successiva)

Elaborazione

- Identificazione dei casi d'uso più importanti (da cui dipende l'architettura)
- Analisi dei requisiti e progettazione
- Costruzione iterativa di versioni del sw sempre più raffinate, in modo da implementare l'architettura globale del sistema
- Identificazione e soluzione dei maggiori rischi

Prodotto: sistema funzionante, per quanto schematico, che fungerà da base per la fase successiva

RUP: fasi (cont.)

Costruzione

- Completamento della progettazione e dell'implementazione del sistema mediante un procedimento iterativo (riscrittura di una parte del codice esistente) e incrementale (a ogni iterazione si aggiungono nuove funzionalità)
- Verifica
- Si riduce il rischio, che spesso deriva dal rimandare i problemi più difficili (quali le fasi di testing e integrazione) alla fine del progetto

Prodotto: sistema da rilasciare

Transizione

- Installazione/distribuzione del prodotto
- β -test
- Formazione degli utenti
- ...

RUP: fasi (cont.)

Tutte le fasi sono iterative, in particolare elaborazione e costruzione

Quando si passa dall'elaborazione alla costruzione?

∃ due risposte:

- nel momento in cui si può adottare una pianificazione predittiva
- quando si è raggiunta una visione generale dei requisiti e dell'architettura del sistema, che non dovrebbe subire grandi mutamenti fino alla fine del progetto

Sviluppo del software: rischi

Rischi	Gestione
<i>Relativi ai requisiti:</i> quali sono i requisiti del sistema? Se questi non sono compresi si costruisce un sistema che non fa quanto richiesto dal cliente	<ul style="list-style-type: none">● Usare i casi d'uso (non dettagliati)● Realizzare il modello del dominio insieme agli esperti (mediante diagrammi delle classi, delle attività e di stato)
<i>Tecnologici:</i> la tecnologia scelta è in grado di svolgere il compito richiesto? Ricordare che i rischi maggiori sono quelli che riguardano l'aggregazione dei vari componenti di un progetto	<ul style="list-style-type: none">● Costruire prototipi per mettere alla prova ogni tecnologia che si intende utilizzare● Provare più strumenti e valutare quale è il più adatto● Prendere tutte le decisioni circa l'architettura del sistema● Aggregare tutti i componenti che si intendono usare

RUP: Elaborazione (cont.)

Rischi	Gestione
<i>Legati alle risorse umane: si può contare sulla squadra e sull'esperienza necessarie?</i>	<ul style="list-style-type: none">● Dedicare tempo alla formazione● Organizzare la formazione in piccoli blocchi, nel momento in cui è necessaria, e metterla subito in pratica● Fare partecipare al progetto (o a ciascuna area specifica dello stesso) un mentore (sviluppatore esperto), oppure● Fare revisionare il progetto ogni due mesi ca.● Organizzare gruppi di lettura
<i>Politico/aziendali: il progetto verrà sostenuto o intralciato?</i>	

RUP: Pianificazione della fase di costruzione (secondo le tecniche della programmazione estrema)

Passo 1: i clienti (cioè le persone che possono stimare il valore commerciale di ogni caso d'uso) suddividono i casi d'uso in tre categorie, a seconda della loro priorità (ovvero, del valore commerciale)

Passo 2: gli sviluppatori suddividono i casi d'uso a seconda del rischio di sviluppo (“alto rischio” = difficile da implementare / non ben compreso)

Passo 3: gli sviluppatori stimano il tempo necessario allo sviluppo di ogni caso d'uso (analisi, progetto, implementazione, testing, integrazione e documentazione) con la precisione di una settimana-uomo

Controllo: se gran parte del tempo corrisponde ai casi d'uso a rischio massimo, ritornare alla fase di elaborazione

RUP: Pianificazione della fasi di costruzione (cont.)

Passo 4: determinare la lunghezza di ogni iterazione (che rimarrà fissa per l'intero progetto) = tempo di sviluppo di un piccolo # di casi d'uso (dipende dal linguaggio usato)

Passo 5: calcolo della quantità di sviluppo che è possibile compiere in una iterazione, tenendo conto del # degli sviluppatori e del carico esterno degli stessi

Passo 6: assegnazione dei casi d'uso alle iterazioni (prima quelli ad alta priorità e/o a maggior rischio)

Passo 7: stesura del piano di consegna, le cui scadenze vanno rispettate (eventualmente si possono posporre dinamicamente dei casi d'uso)