

Applicabilità del Modello procedurale Rational Unified Process (RUP)

Premessa

La metodologia RUP nasce dal superamento della visione tradizionale che vede il completamento di un progetto come una rigida sequenza di attività.

Ad esempio per un progetto software normalmente si considera la sequenza delle seguenti attività:

1. analisi dei requisiti: che ha come input le necessità dell'utente e come output la specifica dei requisiti;
2. la progettazione: che ha come input la specifica dei requisiti e come output la specifica di progettazione;
3. la implementazione (codifica e test): che ha come input la specifica di progettazione e ha come output il codice del progetto;
4. l'integrazione: che ha come input il codice e come output il prodotto software finito.

Negli anni 80 l'ingegneria del software affermava essere questo l'unico approccio corretto al problema. L'esperienza ha messo in evidenza una serie di limiti, per tale approccio, che cade in difetto ad esempio quando:

1. in qualche parte del progetto sono state formulate ipotesi non corrette o non esaustive, che determinano sviluppi successivi non corretti e difficilmente gestibili;
2. alcuni aspetti non tecnici del progetto intervengono a modificare le condizioni di contesto in maniera non del tutto prevedibile a priori – ad esempio via via che il progetto avanza i requisiti dell'utente cambiano perché l'utente stesso ha modo di migliorare la propria conoscenza del problema e quindi di rivedere i precedenti requisiti -;
3. si cerca di sfruttare approcci precedentemente utilizzati, ma in contesti non perfettamente sovrapponibili e/o coerenti, portando il modello al di là dei fisiologici limiti di tolleranza;
4. ecc...

La risposta del RUP – e della ingegneria del software più in generale – al problema della inadeguatezza dei modelli sequenziali è abbastanza ovvia:

- in ogni progetto reale è indispensabile adottare una qualche forma di iterazione che permetta di recuperare tutti quegli aspetti che in prima battuta non è stato possibile cogliere o gestire;
- l'iterazione deve essere rigidamente controllata per evitare di perdere il controllo del progetto;
- devono essere considerati in maniera esplicita i principali rischi in cui può incorrere il progetto e devono essere opportunamente gestiti.

Ogni progettista applica in maniera abbastanza inconsapevole il concetto di iterazione al proprio operato, e il più delle volte in maniera informale: ad esempio, se in fase di stesura del codice di un applicativo ci si rende conto di non aver previsto una determinata funzionalità, a causa di una inadeguata specifica di progettazione, che a sua volta era dovuta ad una inadeguata formulazione dei requisiti, normalmente si introduce la funzionalità mancante senza preoccuparsi di rimettere mano alle specifiche dei requisiti o alle specifiche di progettazione.

Questo porta con se una serie di fattori negativi ben noti:

- disallineamento fra specifiche di vario livello e codice applicativo;
- impossibilità di testare compiutamente l'applicativo da parte dell'utente finale che si troverà di fronte ad una serie di funzionalità, magari indispensabili, ma che non sono direttamente correlate con le necessità espresse in sede di specifica;
- ecc...

In definitiva quindi che una corretta forma di iterazione in progetti di ogni dimensione e di ogni sorta sembra dato acquisito dalla ricerca e in accordo con il buon senso di ogni progettista.

Un ulteriore aspetto da prendere in considerazione è se il RUP nella generalizzazione che opera del processo di progetto sequenziale, in qualche modo ne neghi la natura o ne stravolga la sequenza.

Una possibile risposta può essere data considerando il fatto che il RUP attraverso il concetto di iterazione mette in atto non un processo sequenziale, ma un insieme di processi sequenziali ciascuno dei quali innescato da una nuova iterazione.

Questo punto di vista deve essere recepito, ovviamente, con il buon senso del caso: dire che ogni iterazione debba necessariamente prevedere una analisi dei requisiti non deve necessariamente dire che ad ogni iterazione sia necessario coinvolgere nuovamente gli utenti. Ogni iterazione deve portare a riconsiderare non solo una determinata fase, ma anche tutte quelle a monte, alle quali non sarà necessario rimettere mano se già conformi alle necessità.

Di fatto quindi il RUP riafferma ed estende i contributi che negli anni 80 l'ingegneria del software ha sintetizzato nell'approccio sequenziale, non li nega, tanto da poter affermare che la banalizzazione di un progetto condotto secondo la metodologia RUP può essere un progetto sequenziale nel caso questi sia realizzabile con un'unica iterazione.

Rimane da analizzare una ulteriore questione: in progetti di limitate dimensioni quale deve essere il livello di "cerimonia" adottato, o meglio, quale deve essere il livello di formalismo più idoneo in maniera tale da contemperare l'esigenza del contenimento dei costi di progetto e della qualità dello stesso.

La risposta potrebbe essere molto pragmatica del tipo: QUELLO CHE SERVE. Il RUP tende infatti a favorire buone regole di progettazione attraverso l'imposizione di adeguate propedeuticità alle varie attività, ma consente anche il recupero di inadeguatezze che si rendessero evidenti strada facendo attraverso la messa in campo di opportune iterazioni.

Questo significa che un progetto può partire con il livello di pianificazione e di analisi che si considera adeguato e in corso d'opera può essere riaggiustato se tali ipotesi si dovessero rivelare inadeguate.

Struttura del modello

Secondo la metodologia RUP un progetto può essere visto almeno in due diverse dimensioni:

- il tempo
- le attività.

Secondo l'ottica temporale un progetto attraversa in genere le fasi successive di:

- ideazione;
- elaborazione;
- costruzione;
- transizione.

Le attività che portano al progetto finito possono essere così suddivise:

Attività principali

1. flusso di modellazione di Business
2. flusso dei requisiti
3. flusso analisi e progettazione
4. flusso implementazione
5. flusso test
6. flusso rilascio

Attività di supporto:

1. gestione di progetto
2. gestione delle modifiche e della configurazione
3. infrastruttura (environment)
- 4.

Secondo il RUP le varie tipologie di attività attraversano tutte le fasi di progetto, pertanto non è possibile individuare il punto preciso in cui una attività ha inizio o ha termine. È tuttavia altrettanto vero che, molto pragmaticamente, il RUP afferma esistere una preponderanza di talune attività in talune fasi: ad esempio l'attività di analisi dei requisiti e di sviluppo della architettura hanno sicuramente il loro massimo nella fase di elaborazione del progetto, così come l'attività di implementazione ha il proprio massimo in fase di costruzione e l'attività di test ha il proprio apice nella fase di transizione del progetto.

Verifica, eventualmente a posteriori, della qualità di un progetto

Chiarito ciò rimangono da analizzare ulteriori importanti questioni, secondo l'ottica della commissione informatica. In particolare il problema che mi pongo è: come farà un collega a dimostrare di aver ben operato agli occhi della commissione ?.

In quanto una cosa è aver ben operato, cosa diversa è essere in grado di dimostrarlo agli occhi di un terzo.

La verifica dell'andamento di un progetto o della qualità dello stesso – attività talvolta chiamata monitoraggio – può avere luogo solo qualora esistano i prerequisiti di una tale attività, ovvero occorra esistano gli estremi documentali e le necessarie evidenze. Ad esempio se a posteriori occorre verificare se la pianificazione del progetto è stata ben condotta – la verifica dei tempi e delle risorse impiegate sono tipiche attività di monitoraggio – occorre che il progetto abbia previsto una attività formale di pianificazione che abbia lasciato traccia documentale delle previsioni effettuate e degli scostamenti - la baseline e i vari delta -. Se la pianificazione è stata condotta in termini informali e l'unico elemento certo è, ad esempio, la data contrattuale di fine lavori, nulla si potrà dire in merito alla responsabilità del project manager. Sarà infatti molto complesso, o di fatto impossibile, distinguere le eventuali responsabilità del project manager, da quelle dei vari fornitori o del committente stesso, nel caso esso avesse in carico la predisposizione di determinati prerequisiti.

Affinché un giudizio a posteriori sia formulabile occorre individuare un set minimo di informazioni obbligatorie, su cui la commissione possa condurre la propria valutazione.

Risulta quindi estremamente utile, oserei dire indispensabile, l'elenco di DOCUMENTI predisposto dal gruppo, e risulta a mio parere molto opportuna la graduazione degli stessi in funzione della dimensione del progetto. Quello che a mio modo di vedere è importante chiarire è che tali documenti possono derivare da una attività di progetto, ma sono indispensabili più che al progetto al monitoraggio dello stesso, hanno cioè il valore di evidenza documentale che una certa fase è stata condotta, e quindi costituiscono elemento essenziale per la certificazione della qualità del percorso progettuale.

Ad esempio proviamo a considerare sotto questa ottica un documento quasi sempre presente come il preventivo di fornitura: si supponga che esso consista in una descrizione succinta dell'oggetto della fornitura, comprensivo del costo della stessa dettagliato almeno nelle macro voci principali, comprenda poi la data prevista di inizio attività e la data prevista di fine attività. Si supponga inoltre che il preventivo sia controfirmato oltre che da fornitore anche dal committente.

Dal punto di vista del monitoraggio, la data di inizio attività e la data di fine attività costituiscono una grossolana baseline del progetto, che potrà essere confrontata con l'inizio effettivo delle attività e con la fine – ufficializzata dal collaudo - .

Per cui che il preventivo debba esistere non è esigenza data dalla comune pratica commerciale, ma in questo contesto specifico esigenza fondamentale per consentire il monitoraggio.

Pertanto il gruppo deve chiedersi, quali sono i documenti necessari a rendere possibile l'azione di monitoraggio possibile ?

Ritengo che i documenti che l'ordine debba richiedere siano quelli che scandiscono un tipico progetto sequenziale - se è vero che il RUP nella versione banale coincide con un progetto

sequenziale - comprensivi dei vari aggiornamenti – se è vero che il contributo più importante del RUP è l'iterazione, e che ogni iterazione deve portare ad un incremento verificabile e certificabile del progetto stesso -.

Ad esempio supponiamo che i documenti base da richiedere per certificare lo stato di avanzamento di un progetto siano:

1. Analisi delle necessità dell'utente;
2. Specifica dei requisiti;
3. Specifiche di progetto;
4. Codici sorgenti, eseguibili, configurazioni di ambiente;
5. Test di verifica e relativi output;
6. Specifiche di integrazione;
7. Test di verifica dell'integrazione e relativi output.

Integrati dalla seguente documentazione:

1. Documentazione di project management
2. Documentazione relativa alle modifiche e alla configurazione – si riferisce in particolar modo ai progetti software in cui esistono specifici problemi di versioning e di configurazione, gestibili con strumenti di supporto appositi -
3. infrastruttura (environment) - si riferisce in particolar modo ai progetti software in cui esistono specifici problemi di contesto nel quale deve e può essere calato un determinato prodotto, ad esempio il necessario sistema operativo, le necessarie interfacce applicative, i necessari DBMS, ecc... -.

Ogni documento dovrà riportare la data e ora di prima emissione e la data e ora di ogni successiva modifica e la natura delle modifiche apportate – con le modalità previste dai tipici sistemi qualità -.

Attraverso questi documenti sarà possibile ricostruire la storia del progetto e l'adozione o meno di un percorso progettuale congruo.

PROBLEMA: non è detto che il progettista sia disponibile a fornire ad un eventuale valutatore, o monitoratore, tutto il materiale sopra elencato in quanto coperto da segreto industriale o comunque per diverse ragioni non divulgabile. Ma in assenza di tale materiale viene quindi da chiedersi se sia effettivamente conducibile l'azione di monitoraggio.

Ad esempio: si supponga che il valutatore sia chiamato a decidere se un determinato progetto sia stato correttamente condotto o meno. Come sarà possibile decidere se l'analisi dei requisiti ha o meno preceduto la codifica del codice applicativo senza avere una corretta gestione documentale dei requisiti – con data delle varie versioni – e una corretta gestione del versioning del codice applicativo, con il codice ad una certa data e il codice in versione $X + 1$ alla data successiva ?

Ma la gestione del versioning sul software quante volte viene adottata, nei progetti reali ?

Naturalmente quello sopra esposto è un caso limite, ma secondo me serve a rendere il problema.

Una prima parziale soluzione al problema sopra esposto è quello di fissare dei MILESTONES, peraltro previsti già istituzionalmente dal RUP, nei quali fare le verifiche sopra esposte: ad esempio il fornitore potrà produrre tutte le versioni applicative che vorrà, non soggette a controllo e quindi eventualmente disallineate con la documentazione, ma ad un punto di progetto

concordato tutto dovrà essere allineato e coerente, tale punto di sincronizzazione sarà il MILESTONE X.

I milestones andranno previsti a priori in fase di pianificazione del progetto e andranno concordati con il cliente.

Per progetti particolarmente piccoli potranno esistere anche solo i milestones banali – INIZIO e FINE progetto – e questo significherà che solo alla fine del progetto tutta la documentazione dovrà essere allineata e coerente, ecc... In questo caso, però non potrà essere condotta nessuna attività di monitoraggio delle fasi intermedie...

Una seconda parziale soluzione ai problemi sopra citati – in particolare a quello della non divulgabilità di determinate informazioni coperte da segreto industriale – potrebbe essere quello di definire una serie di informazioni che fungano da indicatore di un determinato aspetto senza tuttavia violare alcun segreto: ad esempio al milestone X dovrebbe essere possibile verificare la corrispondenza fra l'analisi dei requisiti forniti, e il codice prodotto non ispezionando i codici sorgente, ma utilizzando le disponibili funzionalità utente o di sistema compilate, ovvero con un atteggiamento "ai morsetti" o "black box" che dir si voglia.