

Applicazione della gestione requisiti i casi d'uso

Roger Oberg, Leslee
Probasco e Maria Ericsson

White paper di Rational Software

TP 505 (Versione 1.4)

Rational[®]
the software development company

Indice

Sviluppo di software e di sistema nel Periodo di processo1
Perché gestire i requisiti?1
Cos'è un requisito?2
Cos'è la gestione dei requisiti?...	...2
I problemi della gestione dei requisiti2
Skill di gestione dei requisiti3
Skill fondamentale 1: analizzare il problema4
Skill fondamentale 2: comprendere le esigenze dello stakeholder4
Skill fondamentale 3: definire il sistema4
Skill fondamentale 4: gestire l'ambito del sistema...	..5
Skill fondamentale 5: perfezionare la definizione di sistema...	..5
Skill fondamentale 6: gestire la modifica dei requisiti...	..6
Nozioni di requisiti importanti...	.6
Tipi di requisiti...	.7
Team interfunzionali7
Tracciabilità8
Attributi multidimensionali...	...8
Cronologia modifiche...	..10
Attivazione della gestione dei requisiti10
Gestione dei requisiti: dettagli del flusso di lavoro11
Dettaglio del flusso di lavoro: analizzare il problema12
Dettaglio del flusso di lavoro: comprensione delle esigenze dello stakeholder...	...14
Dettaglio del flusso di lavoro: definire il sistema...	..16
Dettaglio del flusso di lavoro: gestire l'ambito del sistema...	...17
Dettaglio del flusso di lavoro: perfezionare la definizione di sistema...	...18
Dettaglio del flusso di lavoro: gestire la modifica dei requisiti19
Riepilogo21
Riferimenti22

Per novizi o dotati di una certa dimestichezza con la gestione dei requisiti e interessati a migliorare il processo dei requisiti, questo documento offre un framework da cui partire per sviluppare un approccio personale.

Casi d'uso e SRS (Software Requirements Specifications)

Per rendere i flussi di lavoro di gestione dei requisiti più significativi ai lettori, gli autori hanno scelto tipi di documento specifici e altri artefatti dall'Unified Process di Rational Software e dal linguaggio di modellazione unificato (UML) standard del settore, i quali consigliano entrambi un processo di progettazione basato su caso d'uso.

Perciò, viene qui descritto un approccio basato su caso d'uso per specificare i requisiti software. Tali flussi di lavoro possono essere utilizzati anche con SRS (ad esempio, standard IEEE)

al posto o in aggiunta dei modelli di caso d'uso.

Sviluppo di software e di sistema nel Periodo di processo

Per la maggior parte dei team di sviluppo di software e di sistema, gli anni novanta sono stati anni relativi a una intensa attenzione relativa al processo, se paragonati a quanto avveniva precedentemente, quando si andava a ruota libera. Sono stati introdotti e diffusi standard di misurazione e certificazione di processo di sviluppo software efficace. Sono stati pubblicati molti libri e articoli inerenti al processo di sviluppo del software e materiale di modellazione e riprogettazione di processo di business. L'aumento degli strumenti software ha agevolato la definizione e l'applicazione del processo di sviluppo efficace del software. La dipendenza dell'economia globale dal software è aumentata durante il decennio, favorendo lo sviluppo dei processi e il miglioramento della qualità di sistema.

Per cui, come si spiega l'elevata incidenza attuale di progetti software non riusciti? Perché esistono molti, se non la maggioranza, progetti software ancora affetti da ritardi, sforamenti di spesa e problemi di qualità? In che modo è possibile migliorare la qualità dei sistemi che vengono creati mentre i nostri affari, le economie nazionali e le attività quotidiane diventano sempre più dipendenti da loro?

Le risposte, come sempre, risiedono nelle persone, negli strumenti e nei processi applicati alla nostra professione. La gestione dei requisiti viene spesso proposta come una soluzione ai problemi crescenti dello sviluppo del software, ma viene ancora prestata poca attenzione al miglioramento della pratica di questa disciplina.

In questo documento vengono presentati gli elementi di un processo efficace di gestione dei requisiti e si evidenziano gli ostacoli a una sua implementazione con successo.

La gestione dei requisiti si applica in modo uguale ai progetti di solo software e ai progetti in cui il software è solo una parte del risultato finale o non è incluso affatto. Per convenienza, da qui in poi si utilizzerà il termine "sistema" per intendere una qualsiasi o tutte queste cose. Comunque, è la natura astratta dello sviluppo del software, da solo o in combinazione con l'hardware, che complica la gestione dei requisiti, l'obiettivo principale di questo documento.

Perché gestire i requisiti?

Detta semplicemente, i team di sviluppo di sistema ritengono di dover gestire i requisiti perché desiderano che i loro progetti abbiano successo. La soddisfazione dei requisiti dei loro progetti ne determina il successo. Una gestione sbagliata dei requisiti diminuisce la probabilità che tali obiettivi vengano raggiunti.

Ciò che accade di recente è di supporto a tale tesi:

- ☐ I report CHAOS di Standish Group che vanno dal 1994 al 1997 hanno stabilito che i contributi più significativi al fallimento dei progetti sono correlati ai requisiti. [1]
- ☐ Nel dicembre del 1997, il Computer Industry Daily ha riportato uno studio di Sequent Computer Systems, Inc. in cui, tra 500 responsabili IT interrogati negli USA e nel Regno Unito, il 76 per cento di coloro che hanno risposto ha avuto esperienze di fallimenti di progetto completi durante la propria carriera. La causa di fallimento principalmente indicata era quella della "modifica dei requisiti dell'utente." [2]

L'eliminazione del rischio di fallimento dovrebbe costituire una motivazione sufficiente per gestire i requisiti. L'aumento delle probabilità di riuscita di un progetto e di altri vantaggi derivanti dalla gestione dei requisiti, può costituire allo stesso modo una valida motivazione. Il report CHAOS di Standish Group ha inoltre stabilito che una buona gestione dei requisiti rappresentava il fattore di principale rilevanza nel successo dei progetti.

Cos'è un requisito?

Il primo passo verso la comprensione della gestione dei requisiti è quello di accordarsi su un vocabolario comune. Rational definisce un requisito come "una condizione o una capacità alla quale il sistema [che viene creato] deve conformarsi". L'IEEE (Institute of Electronics and Electrical Engineers) utilizza una definizione simile.

Gli autori ben conosciuti di progettazione di requisiti Merlin Dorfman e Richard H. Thayer offrono una definizione più compatibile e raffinata che è specifica ma non necessariamente limitata al software.

"Un requisito software può essere definito come:

- ☐ Una capacità software necessaria per l'utente per risolvere un problema o raggiungere un obiettivo.
- ☐ Una capacità software che deve essere soddisfatta o posseduta da un sistema o da un componente del sistema per soddisfare un contratto, una specifica, uno standard o altra documentazione imposta formalmente". [3]

Cos'è la gestione dei requisiti?

Essendo i requisiti degli elementi a cui il sistema in creazione deve conformarsi e poiché la conformità ad alcuni insiemi di requisiti determina il successo o il fallimento del progetto, ha senso scoprire cosa sono i requisiti, prenderne nota, organizzarli e tenerne traccia nel caso si modifichino.

Detto diversamente, la gestione dei requisiti è:

- ☐ un approccio sistematico per ricavare, organizzare e documentare i requisiti del sistema e
- ☐ in processo che stabilisce e mantiene l'accordo tra il cliente e il team del progetto in merito alla modifica dei requisiti del sistema.

Tale definizione è simile alla definizione di "progettazione dei requisiti software" di Thayer e dell'IEEE". La progettazione dei requisiti comprende la raccolta, l'analisi, la specificazione, la verifica e la gestione dei requisiti software, con la "gestione dei requisiti software" con funzione di pianificazione e controllo di tutte queste attività correlate [4]. Tutte queste attività vengono incorporate nella definizione di gestione di requisiti qui presentata e insegnata tramite Rational Software. La differenza risiede principalmente nella scelta della parola "gestione" piuttosto che "progettazione". Il termine gestione costituisce una descrizione più appropriata di tutte le attività coinvolte e mette in risalto in modo accurato l'importanza di tenere traccia delle modifiche per conservare gli accordi tra gli stakeholder e il team del progetto.

Per coloro che non hanno familiarità con il termine "ricavare", viene definito come l'insieme delle attività che i team utilizzano per ricavare o scoprire le richieste dello stakeholder, per determinare le reali esigenze relative alle richieste e per arrivare a un opportuno insieme di requisiti che potrebbe essere associato al sistema per soddisfare tali necessità.

I problemi della gestione dei requisiti

Quali difficoltà potrebbero esserci per un processo il cui scopo è quello di assicurare che un sistema sia conforme alle aspettative per cui è impostato? Le difficoltà vengono alla luce nel momento in cui si mettono in pratica i progetti reali. La figura 1 visualizza i risultati di un'indagine del 1996 tra sviluppatori, responsabili e personale per l'assicurazione della qualità. Mostra la percentuale di coloro che hanno risposto e hanno riscontrato i problemi più frequentemente menzionati relativi ai requisiti.

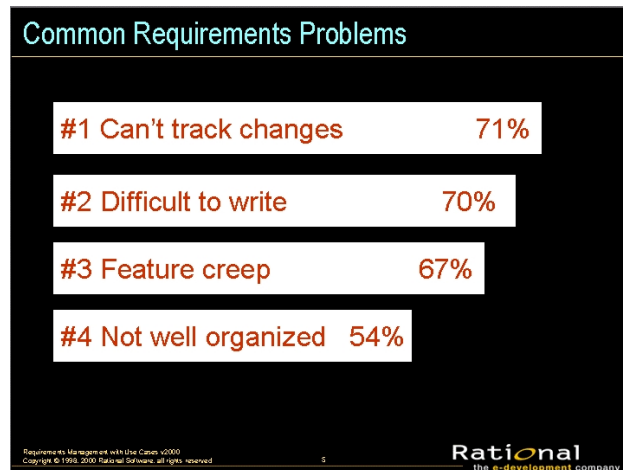


Figura 1: Problemi comuni di requisiti

Un elenco più esauriente di problemi comprende:

- ☐ I requisiti non sono sempre ovvi e possono provenire da molte fonti. ☐ I requisiti non sono sempre facili da esprimere in modo chiaro a parole. ☐ Esistono molti tipi diversi di requisiti nei differenti livelli di dettaglio. ☐ Il numero di requisiti può divenire ingestibile, se non controllato.
- ☐ I requisiti sono correlati gli uni agli altri e ad altri componenti distribuibili del processo in vari modi.
- ☐ I requisiti dispongono di proprietà o di valori di proprietà univoci. Ad esempio, non sono importanti allo stesso modo, né ugualmente facili da soddisfare.
- ☐ Esistono molte parti interessate e responsabili, il che significa che i requisiti devono essere gestiti da gruppi di persone interfunzionali.
- ☐ I requisiti cambiano.
- ☐ I requisiti possono dipendere dal tempo.

Quando questi problemi vengono associati a una gestione di requisiti e a skill di processo inadeguati e alla mancanza di strumenti facili da utilizzare, molti team perdono la speranza di essere in grado di gestire i requisiti in modo efficace. Rational Software ha sviluppato la competenza per istruire i team negli skill e nel processo di gestione dei requisiti. Inoltre, Rational RequisitePro® è uno strumento accessibile per l'automazione efficace della gestione dei requisiti.

Skill di gestione dei requisiti

Per risolvere i problemi menzionati in precedenza, Rational incoraggia lo sviluppo degli skill fondamentali. Tali skill vengono presentati in basso, in ciò che appare un ordine sequenziale, anche se in un processo di gestione dei requisiti vengono applicati continuamente in un ordine diverso. Qui vengono presentati nella sequenza in cui si applicherebbero alla prima iterazione di un nuovo progetto.

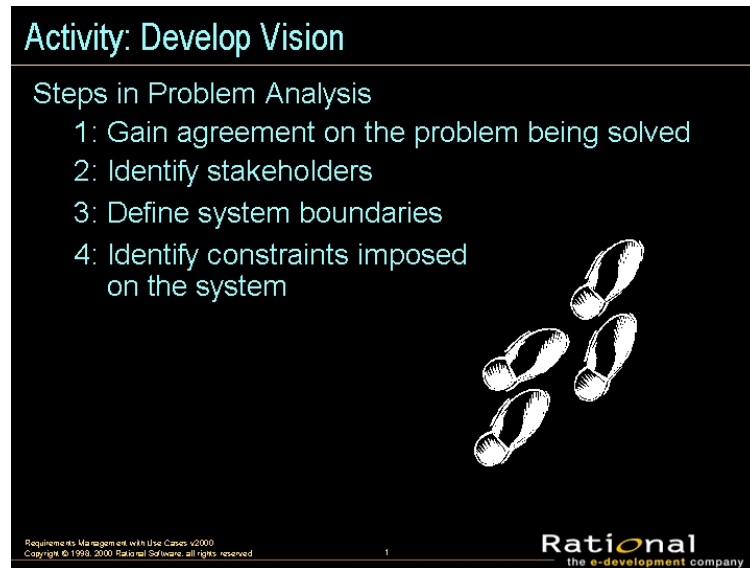


Figura 2: Fasi dell'analisi del problema

Skill fondamentale 1: analizzare il problema

L'analisi dei problemi viene effettuata per capire i problemi di business, le esigenze iniziali degli stakeholder e proporre delle soluzioni ad alto livello. Sono atti di riflessione e analisi per individuare "il problema dietro al problema".

Durante l'analisi dei problemi viene raggiunto un accordo sui problemi reali e vengono identificati gli stakeholder. I limiti e i vincoli della soluzione vengono definiti sia dal punto di vista tecnico che economico. se appropriato, il caso economico per il progetto analizza il ricavo su investimento atteso dal sistema.

Skill fondamentale 2: comprendere le esigenze dello stakeholder

I requisiti possono provenire da molte fonti. Possono provenire da chiunque abbia interesse nell'esito del progetto. I clienti, i partner, gli utenti finali e gli esperti di dominio sono alcune delle fonti di requisiti. I membri del team di progetto e di gestione, le politiche economiche e gli enti normativi possono essere altre.

È importante sapere in che modo determinare quali debbano essere le fonti, ottenerne l'accesso e come dedurre le informazioni da esse. Gli individui che forniscono le fonti principali di queste informazioni vengono definiti gli "stakeholder" del progetto.

Se si sta sviluppando un sistema di informazioni da utilizzare internamente nella propria azienda, si possono includere nel team di sviluppo delle persone con esperienza di utente e nel dominio business. Molto spesso si iniziano le argomentazioni a livello di modello di business, piuttosto che a livello di sistema. Se si sta sviluppando un prodotto che deve essere venduto sul mercato, si possono utilizzare in gran numero delle persone del marketing per una migliore comprensione delle esigenze dei clienti di quel settore di mercato.

Possono verificarsi delle attività di deduzione mediante tecniche quali interviste, brainstorming, creazione di prototipi concettuali, questionari e analisi competitive. Il risultato della deduzione è un elenco di richieste o di esigenze descritte testualmente e graficamente, alle quali è stata assegnata una priorità.

Skill fondamentale 3: definire il sistema

Definire il sistema significa tradurre ed organizzare la comprensione delle esigenze degli stakeholder in una descrizione significativa del sistema da creare. All'inizio della definizione del sistema, le decisioni vengono prese in base a ciò che costituisce un requisito, il formato della documentazione,

la formalità del linguaggio, il grado di specificità dei requisiti, la priorità delle richieste e l'impegno lavorativo stimato, i rischi tecnici e di gestione e

l'ambito. Parte di questa attività può includere dei primi prototipi e dei modelli di progettazione direttamente correlati alle richieste più importanti degli stakeholder.

Viene utilizzata la parola "descrizione" piuttosto che "documento" per evitare la limitazione percepita inerente all'utilizzo comune di quest'ultima parola. Una descrizione può essere un documento scritto, un file elettronico, una immagine o qualsiasi altra rappresentazione intesa a comunicare requisiti di sistema fuorché il sistema stesso.

Il risultato della definizione del sistema è una descrizione del sistema sia grafica che in linguaggio naturale. Nelle successive sezioni verranno forniti alcuni formati consigliati.

Principio 55: scrivere il linguaggio naturale prima di un modello più formale

Se si scrive prima il modello formale, la tendenza sarà quella di scrivere il linguaggio naturale che descrive il modello invece del sistema di soluzione. Considerare i seguenti esempi:

Per effettuare una chiamata su lunga distanza, l'utente deve sollevare il telefono. Il sistema risponderà con un tono di chiamata. L'utente deve premere il tasto "9". Il sistema risponderà con un tono di chiamata diverso...

Il sistema è composta da quattro stati: inattivo, tono di chiamata, tono di chiamata diverso collegato. Per passare dallo stato inattivo a quello di tono di chiamata, alzare il telefono. Per passare dallo stato di tono di chiamata a quello di tono di chiamata diverso, premere il tasto "9".

Si noti come nell'ultimo esempio il testo non aiuti affatto il lettore. — Alan M.

Davis, 201 Principles of Software Development, 1995

Skill fondamentale 4: gestire l'ambito del sistema

L'ambito di un progetto viene definito da un insieme di requisiti su di esso allocati. La gestione dell'ambito del progetto in modo che si adatti alle risorse disponibili (tempo, chiave e denaro) è la chiave di gestione dei progetti di successo. La gestione dell'ambito è un'attività continua che richiede sviluppo iterativo e incrementale, che frantuma l'ambito del progetto in parti gestibili più piccole.

L'uso degli attributi dei requisiti come priorità, sforzo e rischio come base per la negoziazione dell'inclusione di requisiti, è una tecnica particolarmente utile per la gestione dell'ambito. La concentrazione sugli attributi piuttosto che sui requisiti stessi aiuta a desensibilizzare le negoziazioni che sono controverse sotto altri punti di vista.

È inoltre utile per i leader del team essere addestrati agli skill di negoziazione e per il progetto avere un campione nell'organizzazione, così come dal lato del cliente. I campioni di prodotto/progetto devono avere la capacità organizzativa di rifiutare le modifiche dell'ambito oltre le risorse disponibili o di espandere le risorse in modo da accogliere l'ambito aggiuntivo.

Skill fondamentale 5: perfezionare la definizione di sistema

Con una definizione di sistema di alto livello pattuita e un ambito iniziale compreso abbastanza bene, è sia possibile che economico investire risorse in definizioni di sistema perfezionate. Il perfezionamento della definizione di sistema comprende due considerazioni fondamentali: lo sviluppo di descrizioni più dettagliate della definizione di sistema di alto livello e la verifica che il sistema sarà conforme alle esigenze dello stakeholder e si comporti come descritto.

Le descrizioni rappresentano spesso i materiali di riferimento per i team di progetto. Le descrizioni vengono fatte al meglio avendo l'audience in mente. Un errore comune è quello di rappresentare ciò che è complesso da creare con una definizione complessa, in modo particolare quando l'audience non è in condizione o non è disponibile a riflettere a sufficienza per arrivare all'intesa. Questo porta a delle difficoltà nello spiegare lo scopo del progetto del sistema sia all'interno che all'esterno del team di progetto. Spesso potrebbe essere necessario produrre diversi tipi di descrizioni per audience differenti. Questo documento comprende formati consigliati per il linguaggio naturale dettagliato, per il testo formale e per le descrizioni grafiche. Dopo aver stabilito il formato di descrizione, il perfezionamento continua per il ciclo di vita del progetto.

Skill fondamentale 6: gestire la modifica dei requisiti

Non ha alcuna importanza la definizione accurata dei requisiti, poiché verranno modificati. Di fatto, alcune modifiche dei requisiti sono desiderabili! Ciò significa che il proprio team si sta avvalendo dell'opera degli stakeholder. L'introduzione della modifica dei requisiti rappresenta una misura della sensibilità dello stakeholder e della flessibilità operativa del proprio team, due attributi di team che contribuiscono ai progetti di successo. La modifica non rappresenta un male, tranne quando non viene gestita.

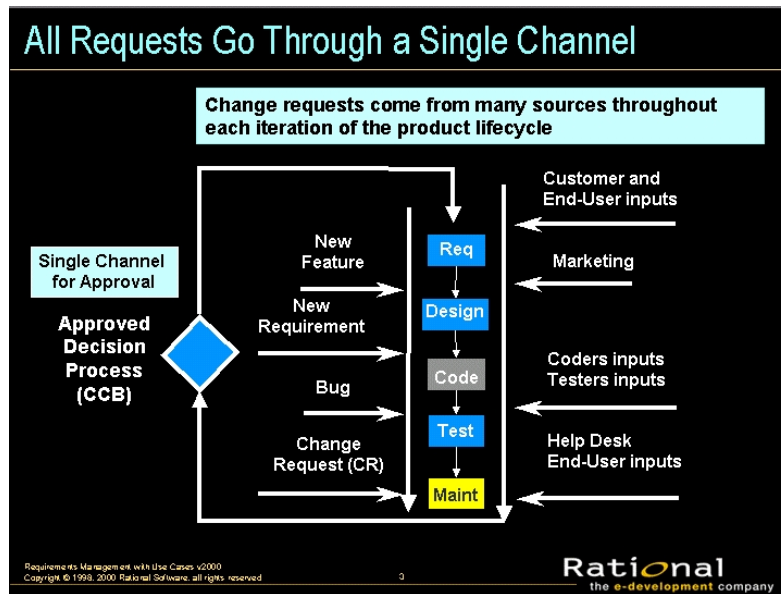


Figura 3: Un processo per la gestione della modifica

Un requisito cambiato significa che è stato impiegato del tempo nell'implementazione di una nuova particolare funzione e la modifica a un requisito può avere un impatto su altri requisiti. La gestione delle modifiche include attività quali stabilire una linea di base, tenere traccia della cronologia di ogni requisito, determinare quali sono le dipendenze di cui è importante tenere traccia, stabilire la tracciabilità fra le voci correlate e mantenere il controllo di versione. Come illustrato nella figura 3, è importante anche stabilire un controllo di modifica o un processo di approvazione, richiedendo la revisione di tutte le modifiche proposte da parte di membri del team designati. Alcune volte, tale canale singolo di controllo di modifica viene chiamato CCB (Change Control Board).

Nozioni di requisiti importanti

Per applicare gli skill di gestione dei requisiti a un progetto, sono utili determinati concetti di gestione dei requisiti affinché chiunque capisca il progetto. Questi comprendono:

- ☐ Tipi di requisiti
- ☐ Team interfunzionali
- ☐ Tracciabilità
- ☐ Attributi multidimensionali
- ☐ Cronologia delle modifiche

Tipi di requisiti

Più un sistema è vasto, più è intricato e maggiore è il numero di tipi di requisiti. Un tipo di requisito è semplicemente una classe di requisiti. Identificando i tipi di requisiti, i team possono organizzare grandi quantità di requisiti in gruppi significativi e più gestibili. Se si stabiliscono tipi differenti di requisiti in un progetto, si consente ai membri del team di classificare le richieste di modifiche e di comunicare più chiaramente.

Di solito, un tipo di requisito può essere diviso o decomposto in altri tipi. Le regole di business e le specifiche della visione possono essere tipi di requisiti di alto livello dai quali i team traggono le esigenze degli utenti, le funzioni e i tipi di requisiti del prodotto. I casi d'uso e altre forme di modellazione guidano i requisiti di progettazione che possono essere decomposti in requisiti software e rappresentati in modelli di analisi e progettazione. I requisiti del test derivano dai requisiti software e vengono divisi in procedure di test specifiche. Quando esistono centinaia, migliaia o persino decine di migliaia di requisiti in un progetto dato, la classificazione dei requisiti in tipi rende il progetto più gestibile.

Team interfunzionali

A differenza di altri processi, quali la verifica o la modellazione di applicazione e che possono essere gestiti all'interno di un singolo gruppo di business, la gestione dei requisiti dovrebbe coinvolgere chiunque può contribuire con la propria esperienza al processo di sviluppo. Dovrebbe includere le persone che rappresentano il cliente e le aspettative di business. Dovrebbero partecipare anche i responsabili di sviluppo, i responsabili di prodotto, gli analisti, i tecnici di sistema e i clienti. I team di requisiti dovrebbero includere anche coloro che creano la soluzione di sistema, cioè i tecnici, gli architetti, i progettisti, i programmatori, gli scrittori tecnici e altri collaboratori tecnici. I tester e altro personale di assicurazione della qualità (QA) dovrebbero essere conteggiati come membri di team importanti.

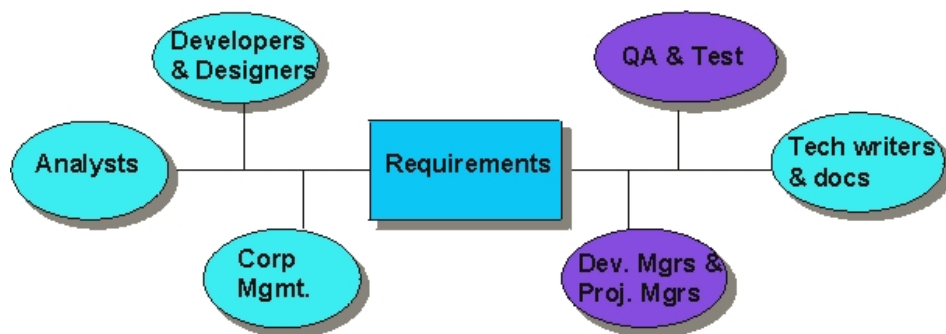


Figura 4: Team di requisiti interfunzionali

Spesso la responsabilità per la preparazione e la manutenzione di un tipo di requisito può essere assegnata da un'area funzionale, per un ulteriore contributo a una migliore gestione di grandi progetti. La natura interfunzionale della gestione dei requisiti rappresenta uno degli aspetti di sfida principali della disciplina.

Tracciabilità

Come suggerito nella descrizione dei tipi di requisiti, nessuna singola espressione di un requisito rimane da sola. Le richieste degli stakeholder sono collegate alle funzioni dei prodotti, proposte per soddisfarle. Le funzioni del prodotto sono correlate ai singoli requisiti che specificano le funzioni in termini di comportamento funzionale e non funzionale. Gli scenari di test sono correlati ai requisiti che essi verificano e convalidano. I requisiti possono dipendere da altri requisiti o essere mutualmente esclusivi. Affinché i team determinino l'impatto delle modifiche e siano certi che il sistema sia conforme alle aspettative, tale relazione di tracciabilità deve essere compresa, documentata e gestita. Anche se la tracciabilità rappresenta una delle nozioni più difficili da implementare nella gestione dei requisiti, è essenziale per assegnare la modifica. Se si stabiliscono tipi di requisiti chiari e si incorpora la partecipazione interfunzionale, è possibile rendere la tracciabilità più facile da implementare e mantenere. Per ulteriori informazioni sulle differenti strategie per la tracciabilità dei requisiti, consultare il white paper, ["Strategie di tracciabilità per la gestione dei requisiti con i casi d'uso" \[5\]](#).

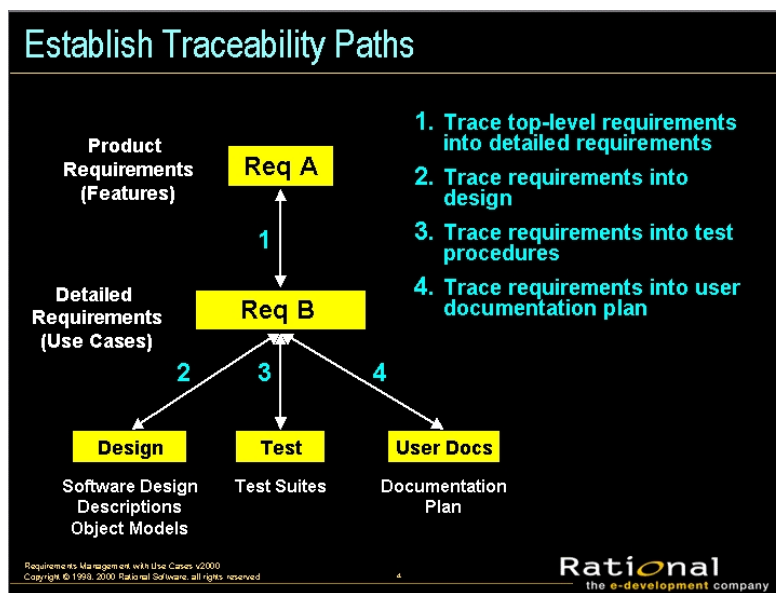


Figura 5: Tracciabilità dei requisiti

Attributi multidimensionali

Ogni tipo di requisito dispone di attributi e ogni singolo requisito dispone di diversi valori di attributo. Ad esempio, è possibile assegnare priorità ai requisiti, identificate da fonti e basi logiche, delegate a specifici team secondari all'interno di un'area funzionale, fornite con un grado difficoltà o associate a una particolare iterazione del sistema. Come illustrazione, la figura 6 mostra gli attributi per un tipo di requisito di funzione da un progetto di apprendimento del tool di gestione requisiti di Rational RequisitePro. come sottinteso dal titolo della schermata, il tipo di requisito e gli attributi per ogni tipo vengono definiti per l'intero progetto, assicurando la coerenza di utilizzo attraverso il team.

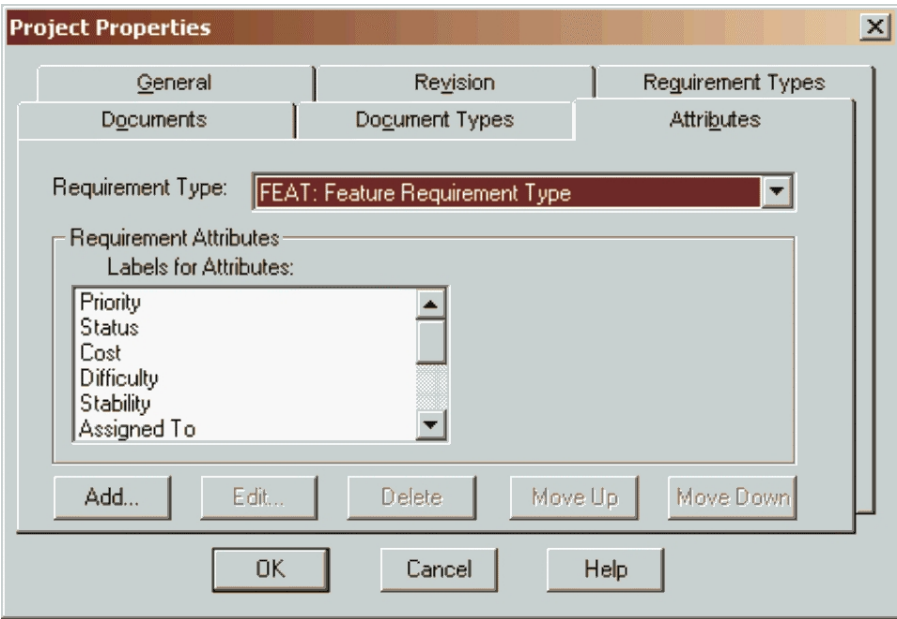


Figura 6: Definizione degli attributi di requisito per le funzioni

In figura 7, vengono mostrate le istanze dei requisiti di funzione per uno specifico progetto in RequisitePro. Si noti come anche senza mostrare l'intero testo per ogni requisito, sia possibile apprendere molto su ogni requisito dai propri valori di attributo. In questo caso, le relative priorità e difficoltà, senza dubbio assegnate da diversi membri del team, agevoleranno il team nell'iniziare a creare l'ambito del progetto per le risorse e il tempo disponibili, tenendo in conto sia le priorità dello stakeholder che un calcolo molto approssimativo dell'impegno riflesso nel valore difficoltà dell'attributo. Nei tipi di requisiti più dettagliati, gli attributi di priorità e impegno possono avere valori più specifici (ad esempio, tempo previsto, linee di codice e così via) con cui perfezionare ulteriormente l'ambito. Questo aspetto multidimensionale di un requisito, composto da diversi tipi di requisiti, ognuno con i propri attributi, è essenziale per l'organizzazione di grandi numeri di requisiti e per la gestione dell'ambito globale del progetto.

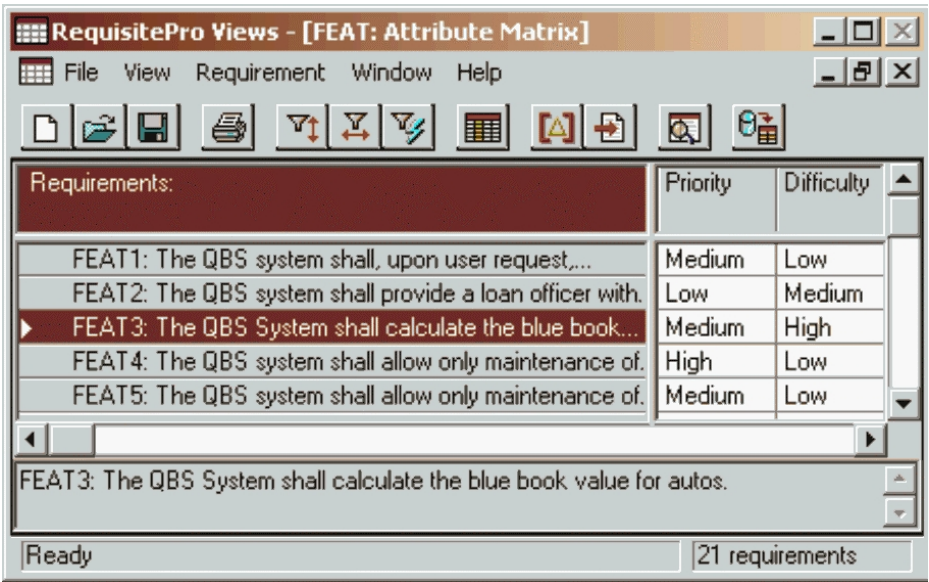


Figura 7: Impostazione di valori di attributo per ogni requisito

Cronologia delle modifiche

Sia un singolo requisito che una raccolta dispongono di cronologie che diventano significative al passare del tempo. La modifica è inevitabile e opportuna per andare di pari passo con un ambiente che cambia e con l'evoluzione della tecnologia. La registrazione delle versioni dei requisiti di progetto consente ai leader del team di acquisire le ragioni per la modifica del progetto, quali un nuovo rilascio di sistema. La comprensione che una raccolta di requisiti può essere associata a una particolare versione di software, permette di gestire la modifica in modo incrementale, riducendo il rischio e migliorando la probabilità di soddisfare i punti fondamentali. Poiché i singoli requisiti si evolvono, è importante comprendere la loro storia: cosa è cambiato, perché, quando e anche per autorizzazione di chi.

Attivazione della gestione dei requisiti

La gestione dei requisiti si occupa degli skill e dei concetti fondamentali presentati in precedenza, per identificare e risolvere i problemi con successo.

Per creare un sistema che soddisfi realmente le esigenze dei clienti, il team di progetto deve prima definire il problema che il sistema deve risolvere. Successivamente, il team deve identificare gli stakeholder da cui ricavare, descrivere e rendere prioritarie le esigenze utente e di business. Occorre concordare un insieme di funzioni di prodotto o di sistema da questo insieme di aspettative o esigenze di alto livello.

I requisiti di software dettagliati devono essere scritti in una forma tale che sia comprensibile sia dai clienti che dal team di sviluppo. Si è accertato che l'utilizzo del linguaggio del cliente nella descrizione di tali requisiti software è più efficace per ottenere la loro comprensione e consenso. Tali requisiti di software dettagliati vengono quindi utilizzati come input per le specifiche di progettazione del sistema, così come per i piani di test e le procedure necessari per l'implementazione e la convalida. I requisiti software dovrebbero anche guidare la pianificazione e la progettazione della documentazione utente iniziale.

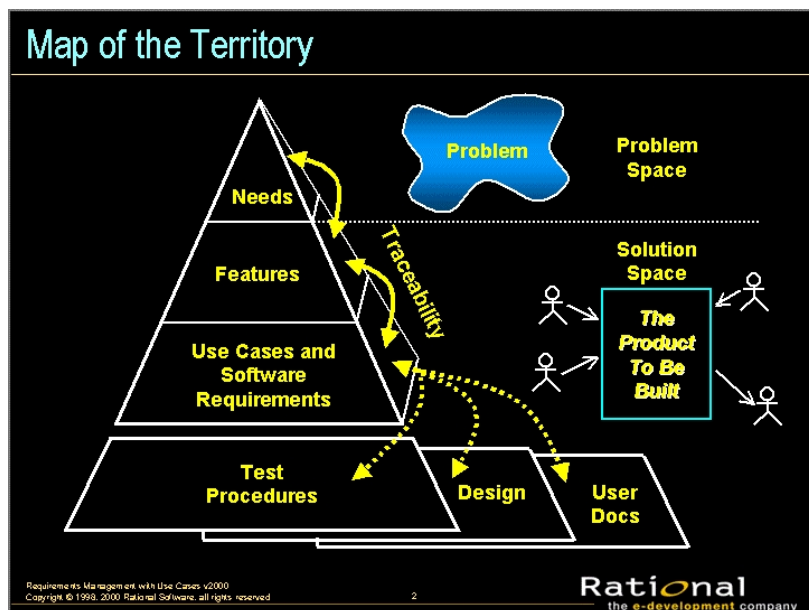


Figura 8: Panoramica della struttura di gestione dei requisiti

Per agevolare ciò, il team di progetto dovrebbe:

- ☐ Accordarsi su un vocabolario comune per il progetto
- ☐ Sviluppare una visione del sistema che descriva il problema che il sistema deve risolvere, così come le relative funzioni principali

- ☐ Ricavare le esigenze degli stakeholder in almeno cinque aree importanti: funzionalità, utilizzabilità, affidabilità, prestazione e supportabilità
- ☐ Determinare quali tipi di requisiti utilizzare
- ☐ Selezionare gli attributi e i valori per ogni tipo di requisito
- ☐ Scegliere i formati in cui vengono descritti i requisiti
- ☐ Identificare i membri che creeranno, contribuiranno o visualizzeranno semplicemente uno o più tipi di requisiti
- ☐ Decidere quale tracciabilità è necessaria
- ☐ Stabilire una procedura per proporre, revisionare e risolvere le modifiche ai requisiti
- ☐ Sviluppare un meccanismo per tracciare la cronologia di requisito
- ☐ Creare report di progresso e di stato per i membri e la gestione del team

Tali attività di gestione di requisiti essenziali non dipendono dal settore, dalla metodologia di sviluppo o dagli strumenti dei requisiti. Sono inoltre flessibili, il che permette una gestione efficace dei requisiti negli ambienti di sviluppo di applicazione maggiormente rigorosi e rapidi.

Alcune parole sui documenti

La decisione di descrivere i requisiti nei documenti merita alcune considerazioni. Da un lato, la scrittura è una forma di comunicazione ampiamente accettata e, per la maggioranza delle persone, un procedimento naturale. Dall'altro lato, l'obiettivo del progetto è quello di produrre un sistema, non documenti.

Il senso e l'esperienza comuni insegnano che la decisione non è se ma come documentare i requisiti. Le maschere da documento forniscono un formato coerente per la gestione dei requisiti. Rational RequisitePro offre tali maschere e la funzione aggiuntiva di collegamento dei requisiti, presenti all'interno di un documento, a un database contenente tutti i requisiti del progetto. Questa funzione unica consente di documentare naturalmente i requisiti mentre vengono resi più accessibili e gestibili in un database relazionale.

Gestione dei requisiti: dettagli del flusso di lavoro

La gestione dei requisiti può seguire un numero infinito di percorsi specifici del dominio. Il seguente approccio descrive sei flussi di lavoro dettagliati che si applicano a ciascuno degli skill fondamentali di gestione dei requisiti, ma che non possono essere applicati a qualsiasi dominio.

I seguenti diagrammi di flusso di lavoro provengono dai dettagli di flusso di lavoro dei requisiti [del Rational Unified Process \[6\]](#). Tali flussi di lavoro vengono espressi in termini di ruoli, attività e artefatti (input o output) come riassunto nella figura 9. Il [testo presente in questo documento](#) descrive brevemente ogni dettaglio di flusso di lavoro, nella speranza di stimolare il pensiero e l'interesse nel migliorare il proprio processo di gestione dei requisiti. È possibile trovare ulteriori informazioni sul Rational Unified Process all'indirizzo www.rational.com.

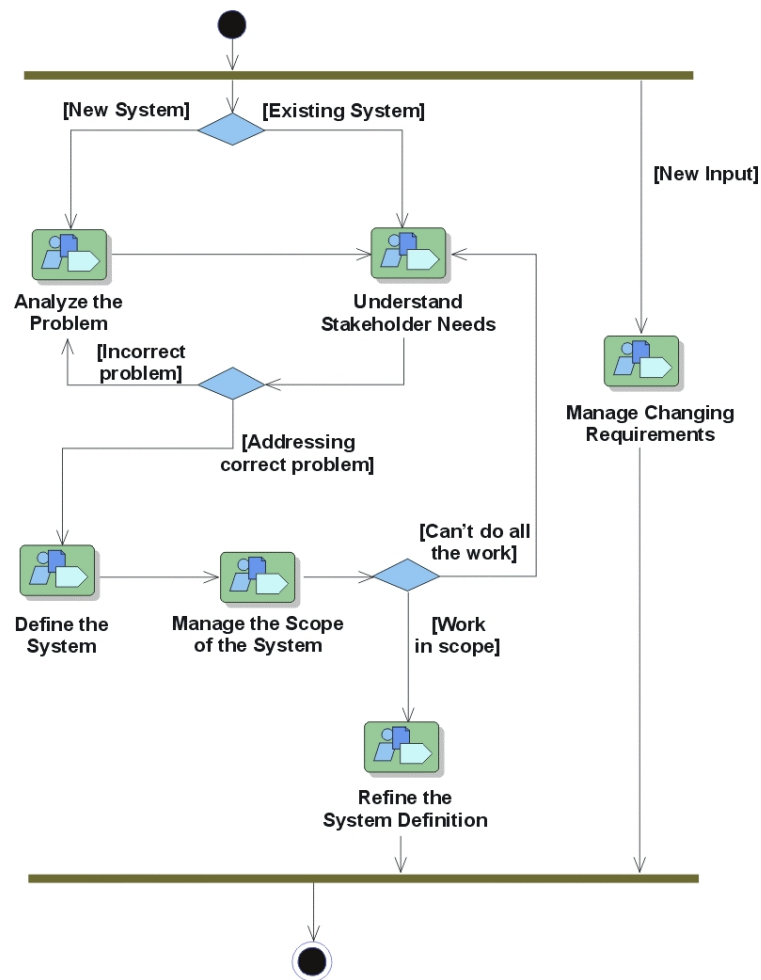


Figura 9: Flusso di lavoro di requisiti nel Rational Unified Process

Dettaglio del flusso di lavoro: analizzare il problema

Nell'analisi del problema, l'attività principale è lo sviluppo della visione per il progetto. L'output proveniente da questa attività è un documento di visione che identifica la vista utente o cliente di alto livello del sistema da creare. La visione esprime i requisiti iniziali come funzioni fondamentali che il sistema deve possedere per risolvere la maggior parte dei problemi critici e soddisfare le esigenze fondamentali dello stakeholder. L'analista di sistema ha il ruolo principale in questo flusso di lavoro. L'analista dovrebbe avere esperienza nel campo dei problemi e una conoscenza del problema; inoltre dovrebbe essere in grado di descrivere un processo che ritiene possa risolvere il problema. Il coinvolgimento attivo di vari stakeholder di progetto è necessario e devono essere considerate tutte le richieste rilevanti di questi ultimi.

Per iniziare l'attività di gestione delle dipendenze, alle funzioni dovrebbero essere assegnati attributi quali fondamento logico, valore relativo o priorità e origine della richiesta. Quando si sviluppa la visione, l'analista identifica gli utenti e i sistemi (gli attori) dei possibili casi d'uso.

Gli attori rappresentano il primo elemento del modello di caso d'uso che definirà i requisiti tecnici funzionali e non funzionali del sistema.

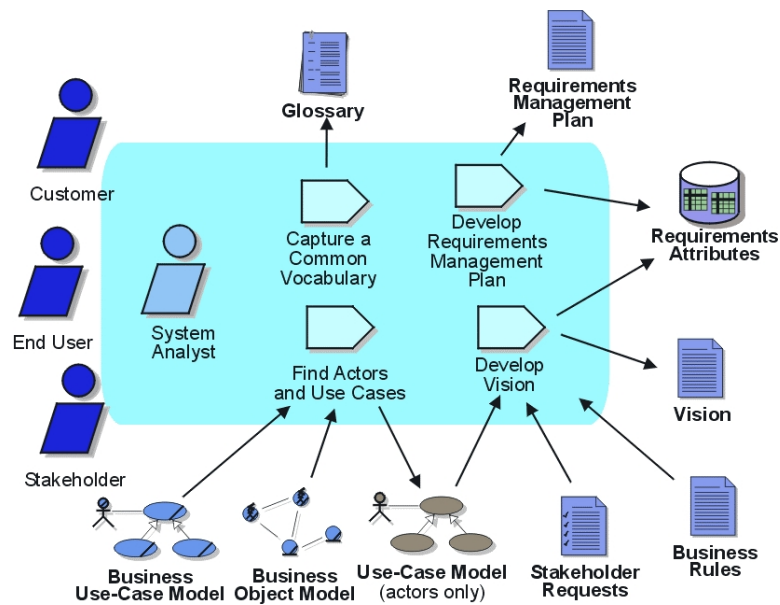


Figura 10: analisi del problema

Dettaglio del flusso di lavoro: analizzare il problema

Iniziazione: uno o più stakeholder che si accorgono di un problema inizieranno il flusso di lavoro.

Un analista di sistema un un team di sviluppo può agevolare una sessione per aiutare gli stakeholder iniziali a descrivere il problema che desiderano risolvere. È importante ottenere un'intesa su una dichiarazione breve del problema percepito. Gli elementi fondamentali di una dichiarazione di problema sono mostrati nella seguente tabella:

Il problema	definizione del problema
Stakeholder interessati	elenco degli stakeholder interessati dal problema
Impatto del problema	descrizione dell'impatto del problema
Soluzione di successo	elenco dei vantaggi fondamentali di una soluzione di successo

La dichiarazione di problema spiega in modo succinto l'obiettivo del progetto. L'analisi del problema stimola ulteriori investigazioni in tutte le richieste dello stakeholder e nel caso di business iniziale, compresi i vantaggi interessanti e una stima approssimativa dei costi. In parallelo alla definizione della dichiarazione di problema, il team dovrebbe compilare anche un glossario per tenere traccia dei termini utilizzati comunemente e accordarsi sulle loro definizioni.

Introduzione del modello di caso d'uso

Un modello di caso d'uso è composto di attori, casi d'uso e loro relazioni. Gli attori rappresentano qualsiasi cosa che deve scambiare informazioni con il sistema, comprendendo quali sono di solito gli utenti chiamati. Quando un attore utilizza il sistema, quest'ultimo esegue un caso d'uso. Un buon caso d'uso è una sequenza di transazioni che produce unrisultato ragionevole di valore per un attore. La raccolta dei casi d'uso è la funzionalità completa del sistema.

—?Jacobson I., Christerson M., Jonsson P., Overgaard G.,Object-Oriented Software Engineering – A Use Case Driven Approach, Addison Wesley - ACM Press, 1992

L'analisi del problema identifica anche i principali attori del sistema. Gli attori sono utenti del sistema o qualsiasi altro sistema che scambia informazioni con esso. In questa fase, l'analisi del problema dovrebbe identificare brevemente alcune modalità ovvie tramite cui l'attore interagirà con il sistema. Le descrizioni dovrebbero essere orientate verso il processo di business piuttosto che il comportamento di sistema. Ad esempio, un programma di creazione budget può consentire a un tipo di attore di "creare budget dipartimentali", laddove un altro attore sarà in grado di "consolidare i budget dipartimentali". L'analista di sistema può successivamente dividerli in casi d'uso aggiuntivi che si adeguano in modo significativo al comportamento di sistema specifico. Ad esempio, "creare budget dipartimentali" potrebbe risultare nei casi d'uso di sistema quali "importare informazioni di fogli elettronici" e "creare viste di budget".

La sessione di analisi di problema descritta sopra viene spesso eseguita più di una volta, probabilmente con diversi stakeholder, e è mischiata con sessioni interne di team di sviluppo. L'analista di sistema che conduce l'incontro con gli stakeholder guiderà una sessione con i membri del team di sviluppo per concepire una soluzione tecnica ai problemi, ricavare le funzioni dagli input iniziali dello stakeholder e abbozzare la descrizione di visione che corrisponde alla prima definizione del sistema da creare. Per agevolare la comprensione della soluzione proposta tra gli stakeholder iniziali, l'analista di sistema può utilizzare gli strumenti di modellazione oppure tecniche di disegno manuale per complementare la descrizione visiva.

Gli stakeholder iniziali vengono consultati in più fasi per aiutare a perfezionare la descrizione del problema e forzare il numero e l'ambito delle soluzioni possibili. Gli stakeholder e gli analisti di sistema gestiscono le dipendenze in questo flusso di lavoro negoziando la priorità delle funzioni principali e ottenendo una comprensione generale delle risorse e dello sforzo necessario per svilupparle. Sebbene si stimi che la priorità e lo sforzo, o la risorsa, siano soggetti a inevitabili modifiche, la gestione delle dipendenze stabilisce all'inizio un importante pattern che resta inalterato durante il ciclo di vita dello sviluppo. Questa è l'essenza della gestione dell'ambito e una previsione iniziale di un successo del progetto.

Dopo diverse stesure, la visione raggiunge un punto in cui il team deve decidere se investire ulteriormente nella deduzione di requisiti. Allo stesso tempo, è stato iniziato separatamente il processo di approvazione del caso di business. Sebbene non trattato più avanti in questo documento, il caso di business descrive tutto ciò che segue:

- ☐ contesto (dominio, mercato e ambito di prodotto)
- ☐ approccio tecnico
- ☐ approccio di gestione (pianificazione, rischio, misura del successo di obiettivo)
- ☐ previsione finanziaria

Dettaglio del flusso di lavoro: comprensione delle esigenze dello stakeholder

Se l'analisi iniziale del problema giustifica l'investimento supplementare, il flusso di lavoro di comprensione delle esigenze dello stakeholder inizia in modo più scrupoloso. L'attività fondamentale è la deduzione delle richieste dello stakeholder. Gli output principali sono costituiti da raccolte di esigenze e funzioni dello stakeholder rese prioritarie, che permettono il perfezionamento del documento di visione e forniscono una comprensione migliore degli attributi di requisiti. Inoltre, durante tale flusso di lavoro è possibile iniziare a discutere sui requisiti di funzionalità del sistema in termini dei suoi casi d'uso e attori. Un altro output importante è rappresentato da un glossario aggiornato dei termini per agevolare il vocabolario in comune tra i membri del team.

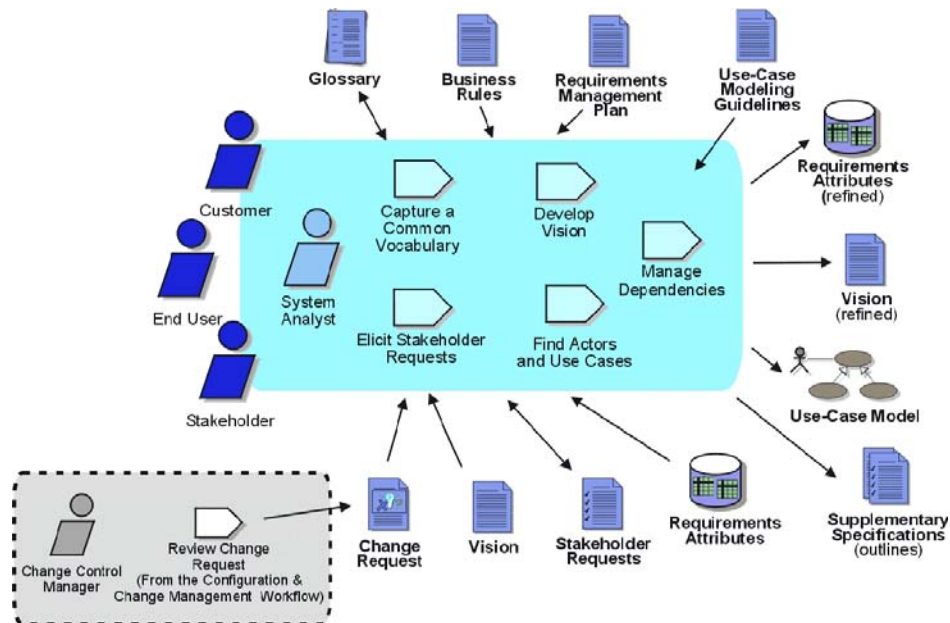


Figura 11: Comprensione delle esigenze degli stakeholder

Dettaglio del flusso di lavoro: comprensione delle esigenze dello stakeholder

L'analista di sistema e gli stakeholder principali identificano ulteriori stakeholder, deducono le loro richieste e determinano le esigenze e le funzioni principali tramite interviste, seminari, storyboard, casi d'uso di processo business e altre tecniche. Uno o più analisti di sistema facilitano tali sessioni. Tali seminari di requisito sono tra le più utili tecniche di deduzione. Il processo comprende gli utenti, il personale di help-desk, i proprietari aziendali, i tester e altri che hanno interesse nel risultato del progetto proposto. Le richieste dello stakeholder sono spesso ambigue, sovrapposte e perfino strane. In aggiunta ai risultati di deduzione formale, le richieste dello stakeholder possono essere espresse in documenti ben formattati, richieste di diminuzione e aumento di database o thread di email e groupware. Gli analisti di sistema registrano, classificano e rendono prioritarie le esigenze principali dello stakeholder identificate.

Comprensione delle esigenze dello stakeholder: dove inizia il "deliziare i clienti"

Le richieste dello stakeholder vengono acquisite quanto più possibile nel linguaggio e formato dello stakeholder richiedente. A differenza dei conseguenti tipi di requisito che vengono solitamente creati da membri di team istruiti e tecnicamente competenti, le richieste di stakeholder sono spesso espresse in modo insufficiente. Sono duplicate o sovrapposte. Possono essere espresse su qualsiasi cosa, da errori di carta a database di richiesta di aumento.

L'analista (o il team che riveste tale ruolo) deve revisionare tutte le richieste, interpretarle, raggrupparle, forse ridigitarle (senza riscriverle) e tradurle in esigenze fondamentali per lo stakeholder e funzioni di sistema nel documento di visione. In base al rigore applicato nello sviluppo e alla disponibilità di tool, è possibile applicare la tracciabilità per alcune o tutte le richieste dello stakeholder, esigenze e funzioni per agevolare la comprensione da parte degli stakeholder di come le loro richieste e necessità vengono considerate nella determinazione dei requisiti del sistema.

La dimostrazione di un serio interesse per la deduzione delle richieste e la soddisfazione delle esigenze dello stakeholder applicando il flusso di lavoro di comprensione delle esigenze dello stakeholder, possono costituire un fattore critico nello stabilire la fiducia dello stakeholder nelle capacità del team di sviluppo.

Basandosi su una migliore comprensione delle esigenze dello stakeholder, l'analista di sistema del team di sviluppo deve perfezionare il documento di visione, prestando attenzione particolare allo sviluppo di una "dichiarazione di posizione di prodotto". In due o tre frasi, questa dichiarazione stabilisce il valore convincente del progetto. La dichiarazione dovrebbe comprendere gli utenti designati, i problemi che risolve, i

vantaggi che apporta e i concorrenti che sostituisce. Tutti i membri del team dovrebbero comprendere tale tema di progetto. Gli analisti di sistema aggiornano anche il glossario per agevolare una comprensione comune dei termini.

Gli stakeholder principali vengono consultati in più fasi per negoziare le priorità delle nuove funzioni derivate dalla comprensione delle esigenze dello stakeholder e per ottenere una comprensione attuale delle risorse e dello sforzo necessario per svilupparle. Come con l'analisi del problema, la gestione delle dipendenze in questo flusso di lavoro aiuta a gestire l'ambito. Stabilisce anche la tracciabilità per le richieste dello stakeholder, le esigenze e le funzioni di sistema, così che gli stakeholder possano assicurarsi che i loro input sono stati considerati.

Dettaglio del flusso di lavoro: definire il sistema

I primi due dettagli del flusso di lavoro dei requisiti, analizzare il problema e comprendere le esigenze dello stakeholder, creano all'inizio iterazioni di definizioni fondamentali di sistema, che includono le funzioni specificate nel documento di visione, un primo profilo del modello di caso d'uso e gli attributi di requisito iniziali. Il successivo dettaglio del flusso di lavoro, definire il sistema, completa la descrizione dei requisiti di sistema di alto livello con il perfezionamento delle definizioni di funzione, l'aggiunta di nuovi attori, casi d'uso e specifiche supplementari.

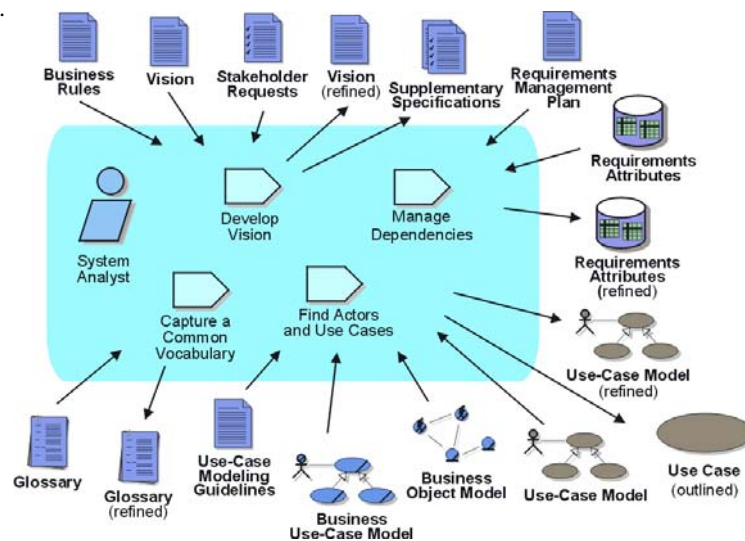


Figura 12: Definizione del sistema

Dettaglio del flusso di lavoro: definire il sistema

Il glossario viene aggiornato per riflettere la comprensione attuale dei termini utilizzati per descrivere le funzioni e i requisiti acquisiti nel modello di caso d'uso e le specifiche supplementari.

L'analista di sistema utilizza l'insieme di funzioni definito nella visione perfezionata per dedurre e descrivere i casi d'uso che elaborano la vista degli utenti del comportamento previsto del sistema. Il modello di caso d'uso serve come contratto tra il cliente, gli utenti e gli sviluppatori di sistema su come funzioneranno le funzioni selezionate nel sistema. Aiuta gli sviluppatori di sistema nel fissare le aspettative e gli obiettivi realistiche aiuta i clienti e gli utenti ad avvalorare che il sistema soddisfa tali aspettative.

Alcuni requisiti di sistema non si adattano bene ai casi d'uso. L'analista di sistema li descrive in una specificazione supplementare. Molti requisiti non funzionali, quali i requisiti di utilizzabilità, di affidabilità, di prestazione e di supportabilità, spesso terminano qui. Si dovrebbe notare che molti requisiti non funzionali di tali tipi sono specifici di un singolo caso d'uso. È meglio per chi specifica il caso d'uso collocare tali requisiti nella stessa specifica del caso d'uso (consultare Perfezionamento del flusso di lavoro di sistema), lasciando la specifica supplementare per i requisiti non funzionali di grandi sistemi.

In questo dettaglio di flusso di lavoro, l'analista di sistema crea e imposta gli attributi per i requisiti supplementari (quali priorità e casi d'uso correlati). Inoltre, l'analista aggiunge e aggiorna i valori di attributo per i casi d'uso iniziali e nuovi.

Infine, l'analista gestisce le dipendenze tenendo traccia delle esigenze fondamentali dell'utente e le funzioni critiche correlate ai casi d'uso e ai requisiti descritti nelle specifiche supplementari.

Dettaglio del flusso di lavoro: gestire l'ambito del sistema

Dopo aver identificato i requisiti a livello di funzione, descrivendo più attori, casi d'uso e altri requisiti specificati nelle specifiche supplementari, l'analista dovrebbe dedurre e assegnare i valori nel modo più accurato possibile agli attributi di requisito come priorità, sforzo, costo e rischio. Ciò consente una migliore comprensione su come determinare l'ambito iniziale del rilascio di sistema e può anche consentire all'architetto di identificare i casi d'uso significativi dal punto di vista dell'architettura.

Il piano di iterazione, sviluppato in parallelo dalla gestione di progetto e sviluppo, appare prima nel dettaglio di questo flusso di lavoro: gestire l'ambito del sistema. Conosciuto anche come piano di sviluppo, il piano di iterazione definisce il numero e la frequenza delle iterazioni pianificate per il rilascio. Gli elementi di rischio maggiori all'interno dell'ambito dovrebbero essere pianificati per le iterazioni iniziali.

Altri importanti output provenienti dal flusso di lavoro di gestione dell'ambito, comprendono l'iterazione iniziale del documento di architettura software e una visione revisionata che riflette la comprensione aumentata degli analisti e degli stakeholder principali della funzionalità del sistema e delle risorse del progetto.

Come per il caso business, menzionato in precedenza e il primo problema del piano di iterazione, il documento di architettura software non è un artefatto dei flussi di lavoro di gestione dei requisiti, sebbene sia correlato e faccia parte del Rational Unified Process. Non è argomento di questo documento.

L'esperienza insegna che i punti importanti per la gestione di successo dell'ambito sono costituiti da una considerazione adeguata dei valori di attributo assegnati alle esigenze dello stakeholder, ai casi d'uso e ad altri requisiti indicati nelle specifiche supplementari; insieme all'interazione regolare, aperta ed onesta con gli stakeholder rappresentativi.

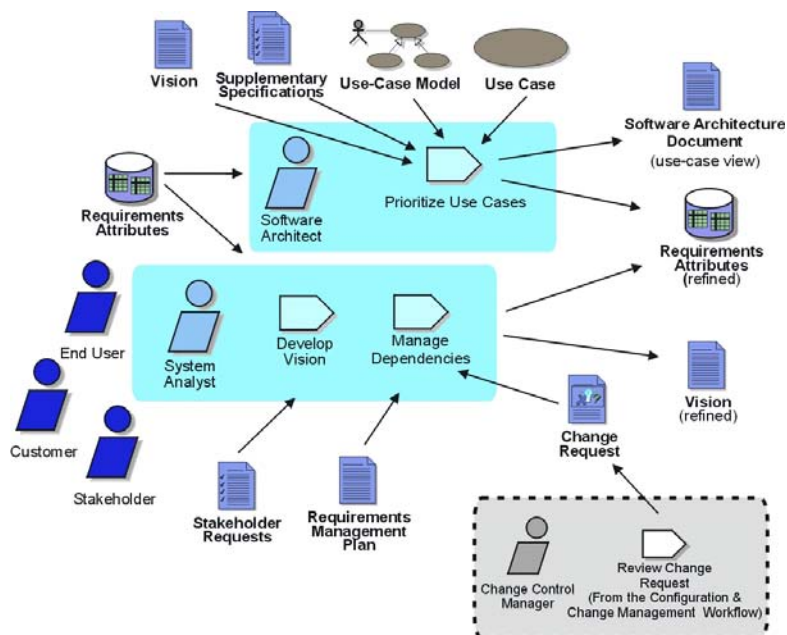


Figura 13: Gestione dell'ambito del sistema