

# Strategie di tracciabilità per la gestione dei requisiti con i casi d'uso

Ian Spence e Leslee Probasco

White paper di Rational Software

TP 166, 1998

---

## Tabella dei contenuti

Sommario ...	...1
Introduzione e background...	...1
Elementi di tracciabilità ...	...1
Tracciabilità esplicita ed implicita...	...1
Gestione degli artefatti di supporto ...	...4
Possibili strategie di tracciabilità...	...4
Perché adottare approcci ibridi? ...	.5
Informazioni relative al catalogo di strategia della tracciabilità ...	..7
Catalogo di strategia della tracciabilità ...	..8
Annotazione diagrammi ...	.8
Tipi di tracciabilità di supporto ...	.9
Nessun modello di caso d'uso ...	...11
Modello di caso d'uso unico ...	...13
Il modello di caso d'uso definisce le funzioni del prodotto ...	...16
Le caratteristiche guidano il modello di caso d'uso ...	...18
Il modello di caso d'uso è un'interpretazione della specifica dei requisiti software ...	...22
Il modello di caso d'uso riconcilia più insiemi di requisiti software tradizionali ...	..27

## Sommario

---

In molte delle applicazioni delle tecniche di modellazione di casi d'uso in commercio il modello del caso d'uso deve essere abbinato a tecniche di cattura requisiti più tradizionali per offrire un processo di gestione dei requisiti accettabile da tutti gli stakeholder coinvolti nel progetto. Questo documento esplora le strategie di tracciabilità disponibili per organizzazioni che adottano tecniche di modellazione di casi d'uso come parte della loro strategia di gestione dei requisiti.

## Introduzione e background

---

### Elementi di tracciabilità

Un punto comune che genera confusione durante la discussione sulla gestione dei requisiti, specie con tool come RequisitePro, è il peso eccessivo dato al termine “requisito”. Oltre agli elementi comunemente definiti “requisiti”, è necessario acquisire e tracciare molti altri tipi di elementi. Tra questi sono incluse problematiche, assunzioni, richieste, termini di glossario, attori, test, ecc...

Catturare e tracciare questi altri elementi consente una gestione più efficace dei requisiti del progetto.

#### Definizione: elemento di tracciabilità

Qualsiasi elemento testuale o di modello necessiti di essere esplicitamente tracciato da un altro elemento testuale o di modello per poter tenere traccia delle dipendenze fra loro.

Nel rispetto di RequisitePro, questa definizione può essere riformulata come:

Ogni elemento testuale o di modello rappresentato in RequisitePro da un'istanza di tipo requisito RequisitePro.

RequisitePro stesso fornisce un eccellente tool per definire, catturare e tracciare i valori, gli attributi ed i link alla tracciabilità tra i diversi tipi di elementi coinvolti nello sviluppo software.

### Tracciabilità esplicita ed implicita

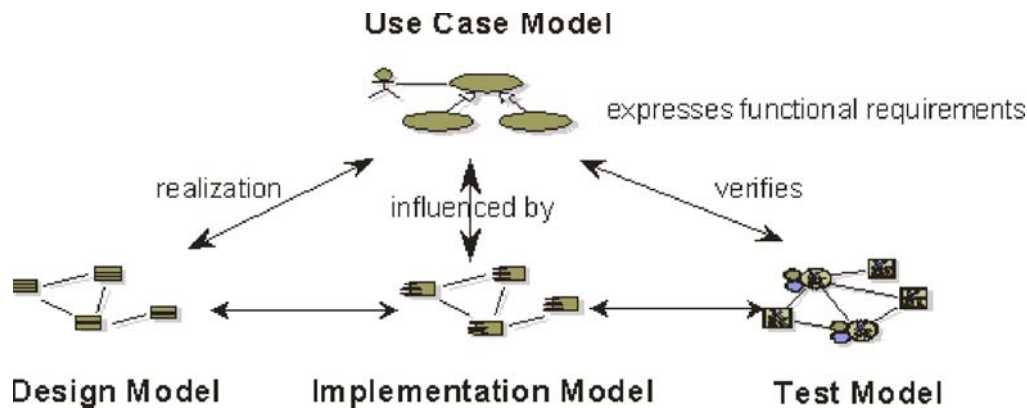
Esiste una certa quantità di tracciabilità implicita in ogni processo di sviluppo. Viene generalmente fornita dalle relazioni formali tra gli artefatti nel processo.

Esempi di questo tipo di tracciabilità sono:

- ☐ Convenzioni di denominazione ossia, la classe nel modello di progettazione denominata Fred viene implementata dalla classe nel modello d'implementazione denominata Fred.
- ☐ La costruzione di associazioni tra modelli ossia, la vista componenti in Rose consente l'associazione dei pacchetti e delle classi nella vista logica in Rose ai pacchetti nel modello d'implementazione. Quest'associazione può contenere una nuova denominazione tra i modelli e l'applicazione di differenti strategie di imballaggio.
- ☐ Le relazioni tra gli stessi elementi di modello ossia, in Rational Unified Process le realizzazioni di caso d'uso nel modello di progettazione vengono tracciate indietro fino ai casi d'uso che realizzano.
- ☐ La creazione di differenti prospettive che illustrano come gli elementi di un modello soddisfino le richieste implicite negli elementi di un altro modello ossia, le realizzazioni di caso d'uso nel modello di progettazione dimostrano come i relativi elementi di modello collaborano per soddisfare il caso d'uso. Queste forniscono una prospettiva nel modello di progettazione, che convalida e allega lo statico imballaggio delle classi ed i pacchetti nel modello di progettazione.

Tutti questi esempi forniscono un livello di tracciabilità e consentono alle analisi d'impatto di essere comprensibili attraverso le informazioni contenute nei modelli di sviluppo.

Come mostrato nella figura sottostante, lo sviluppo guidato del caso d'uso coinvolge una serie di modelli correlati.



La figura mostra i modelli e le loro implicite relazioni. La relazione tra modelli fornisce un livello di tracciabilità implicito al processo di sviluppo.

Se viene intrapreso uno sviluppo guidato di caso d'uso sono richiesti alcuni artefatti a supporto del modello di caso d'uso e per abilitare la definizione di una specifica dei requisiti software completa. In Rational Unified Process questi sono le Specifiche supplementari ed il Glossario. Sono interessanti anche i documenti di Visione e di caso business, che contengono le definizioni delle necessità, degli obiettivi e delle caratteristiche del progetto.

Le relazioni tra modelli non coinvolgono questi artefatti di supporto e dunque non sono coperti dalla implicita tracciabilità costruita nel processo di sviluppo.

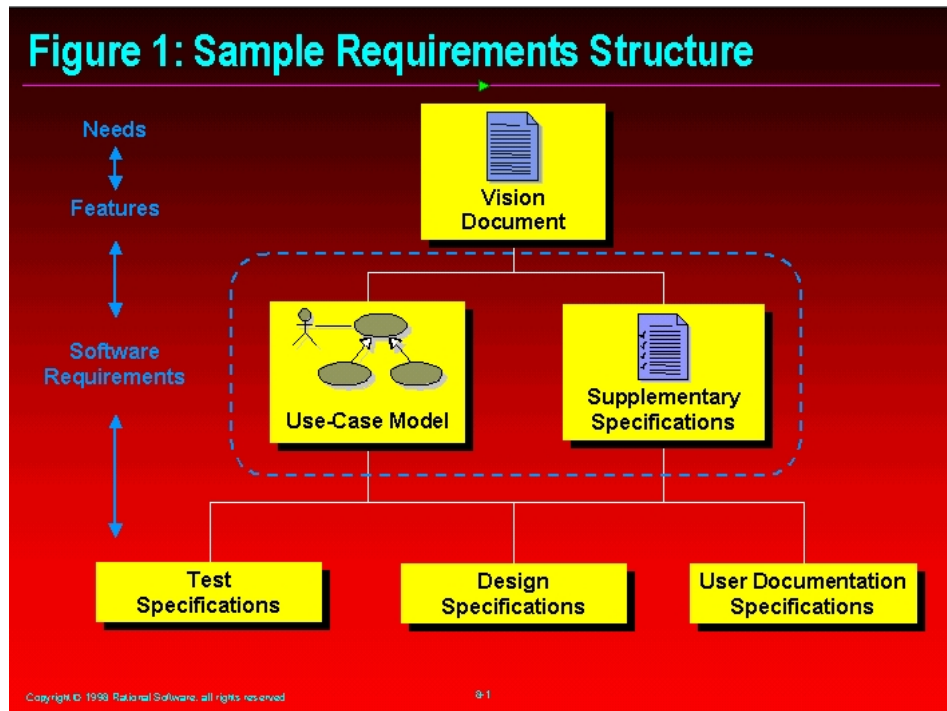
Queste relazioni implicite sono fondamentali per il processo di sviluppo e beneficiano dall'essere costruite come parte naturale del lavoro dello sviluppatore. Sono centrali nel processo di modellazione e vengono costruite e mantenute nel momento in cui i modelli sono maturi.

La tracciabilità implicita è limitata alle relazioni disponibili nelle annotazioni di modellazione.

Dunque, per quale motivo può essere desiderabile un'aggiuntiva tracciabilità esplicita?

Se vengono adottati i principi di tracciabilità dei requisiti è necessario integrare i requisiti, gli elementi di modello ed altri elementi di tracciabilità nella relativa gerarchia. Si può anche preferire di aggiungere relazioni di tracciabilità aggiuntive nel processo di sviluppo.

La seguente figura mostra una gerarchia di tracciabilità di esempio con la relazione tra le "caratteristiche" definite in un documento di visione e i "requisiti software" definiti in un modello di caso d'uso e la specifica supplementare. Mostra inoltre come i requisiti software vengono tracciati nei requisiti di test, nella progettazione e nella documentazione per l'utente.



Analizzando la relazione tra i requisiti supplementari (definiti nel documento delle specifiche supplementari) ed i modelli d'implementazione, si nota che questa non viene coperta dalla tracciabilità implicita tra i modelli.

Questo è un buon esempio di un livello aggiuntivo di tracciabilità esplicita spesso richiesta su un progetto. Molti requisiti tracciano lungo le intere serie di modelli a causa della loro implicita relazione con il modello di caso d'uso. I requisiti supplementari, catturati lungo il modello di caso d'uso nella specifica supplementare, non sono direttamente collegati a nessuno dei pacchetti nel modello di progettazione che li devono considerare, né ad alcun componente in quello d'implementazione che li deve soddisfare.

Altri esempi includono relazioni tra:

- ☐ le caratteristiche del sistema ed il modello di caso d'uso
- ☐ il modello di caso d'uso e la documentazione dell'utente
- ☐ il modello di caso d'uso e i requisiti di test

Una delle principali decisioni da prendere quando viene impostato il processo di tracciabilità dei requisiti consiste nel relativo livello richiesto e nella quantità di tracciabilità esplicita necessaria per conseguire questo obiettivo. Lo scopo è di fornire un approccio alla gestione dei requisiti e della tracciabilità per facilitare il processo di sviluppo e non per complicarlo o restringerlo.

L'aggiunta di tracciabilità esplicita agli artefatti di sviluppo può comportare costi significativi al progetto. È particolarmente importante se vengono considerati i costi a lungo termine per la popolazione ed il mantenimento di queste informazioni aggiuntive. È necessario stabilire un livello di tracciabilità adeguato al progetto ed ottenere un ritorno di investimento per ogni tracciabilità esplicita aggiuntiva che si intende conservare. È preferibile che gli sviluppatori passino il proprio tempo nello sviluppo e non nella traccia. A tal scopo è necessario stabilire e valutare una strategia di tracciabilità prima di aggiungere i costi di tracciabilità esplicita al progetto.

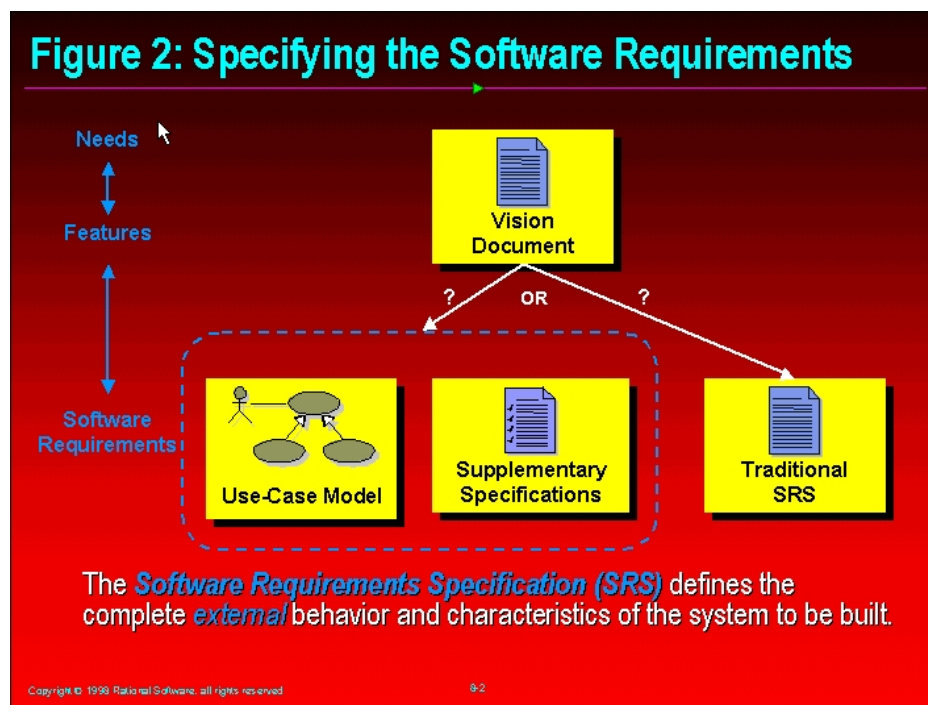
La strategia definisce il livello di tracciabilità esplicita che si desidera aggiungere al processo di sviluppo software.

### Gestione degli artefatti di supporto

La figura 1 mostra gli artefatti coinvolti nella specifica dei requisiti in RUP.

Da notare qui che il modello di caso e la specifica supplementare forma la completa SRS (Software Requirements Specification). Significa che non è necessario avere un documento SRS (Software Requirements Specification) rigoroso come richiesto dalle tecniche di gestione dei requisiti tradizionali.

La figura 2 mostra come un documento SRS tradizionale venga spesso messo in relazione agli artefatti di RUP. L'SRS è solo un modo alternativo di documentare i requisiti software. È importante comprendere che entrambi gli approcci possono fornire una SRS (Software Requirements Specification) che definisce il comportamento esterno completo del sistema da costruire.



Questa relazione è spesso fraintesa poiché si presume che i due modelli di gestione dei requisiti non possano coesistere. Spesso si pensa di dover scegliere tra le tecniche di gestione dei requisiti tradizionali, utilizzando un documento SRS (Software Requirement Specification) formale e le tecniche di gestione dei requisiti basati sul modello di caso d'uso, mediante un modello di caso d'uso ed una specifica supplementare. Di certo in certe circostanze è necessario che le due forme di SRS (Software Requirements Specification) coesistano all'interno dello stesso progetto.

### Possibili strategie di tracciabilità

Esiste più di una strategia di tracciabilità accettabile?

Possono essere utilizzate molte strategie per facilitare il processo di gestione dei requisiti; anche lavorando all'interno del framework di Rational Unified Process sono possibili diversi approcci. Quattro tra i più comuni, tutti visti in azione dagli autori, sono:

- Solo il modello di caso d'uso.

In questo caso il modello di caso d'uso è l'unica istruzione dei requisiti dei sistemi. I progetti che scelgono questo approccio sono caratterizzati da una intima associazione e fiducia tra cliente e sviluppatore.

Il modello di caso d'uso, il glossario e le specifiche supplementari formano l'intera istruzione dei requisiti di sistema - non ci sono ulteriori definizioni di necessità, funzioni del prodotto o requisiti software.

□ Le caratteristiche guidano il modello di caso d'uso

È la strategia predefinita consigliata da Rational Unified Process. Il modello di caso d'uso e le specifiche supplementari formano una specifica di requisiti software completa. Le caratteristiche vengono documentate nel documento di Visione e vengono tracciate verso i casi d'uso. Se non sono riflesse nel modello di caso d'uso allora vengono tracciate verso i requisiti supplementari nelle specifiche supplementari.

In questo caso il modello di caso d'uso agisce come istruzione principale dei requisiti funzionali. Il modello di caso d'uso ed i requisiti supplementari sono integrati con le necessità e le funzioni del prodotto in aggiunta al glossario ed ai requisiti supplementari.

□ Il modello di caso d'uso è un'interpretazione della specifica dei requisiti software

In questo caso il modello di caso d'uso è un'interpretazione di una specifica dei requisiti software tradizionale e rigorosa. Questa viene più spesso utilizzato se tale specifica è un mandato di un protocollo interno o regolatore per cui è richiesto un modello di caso d'uso per abilitare la procedura di sviluppo guidato del caso d'uso.

Le caratteristiche tracciano verso un documento SRS (Software Requirement Specification) (come nella tradizionale caso d'uso. Model.

□ Tale modello riconcilia più serie di requisiti software tradizionali.

Il modello di caso d'uso è l'interpretazione di una serie di SRS formali da fonti multiple e fornisce la specifica di un singolo sistema comune.

In questo caso ogni stakeholder ha la propria serie di funzioni del prodotto e requisiti software, che sono spiegati in dettaglio all'interno dei propri documenti SRS e di Visione. Questi diversi punti di vista e desideri potenzialmente conflittuali vengono poi riconciliati all'interno di un singolo modello di caso d'uso, che specifica cosa il sistema farà. Questa strategia può essere molto efficace nei confronti di una grande quantità di stakeholder indipendenti.

In tutti i casi tranne che per l'opzione 1 il modello di caso d'uso viene combinato con elementi del processo di tracciabilità dei requisiti tradizionale.

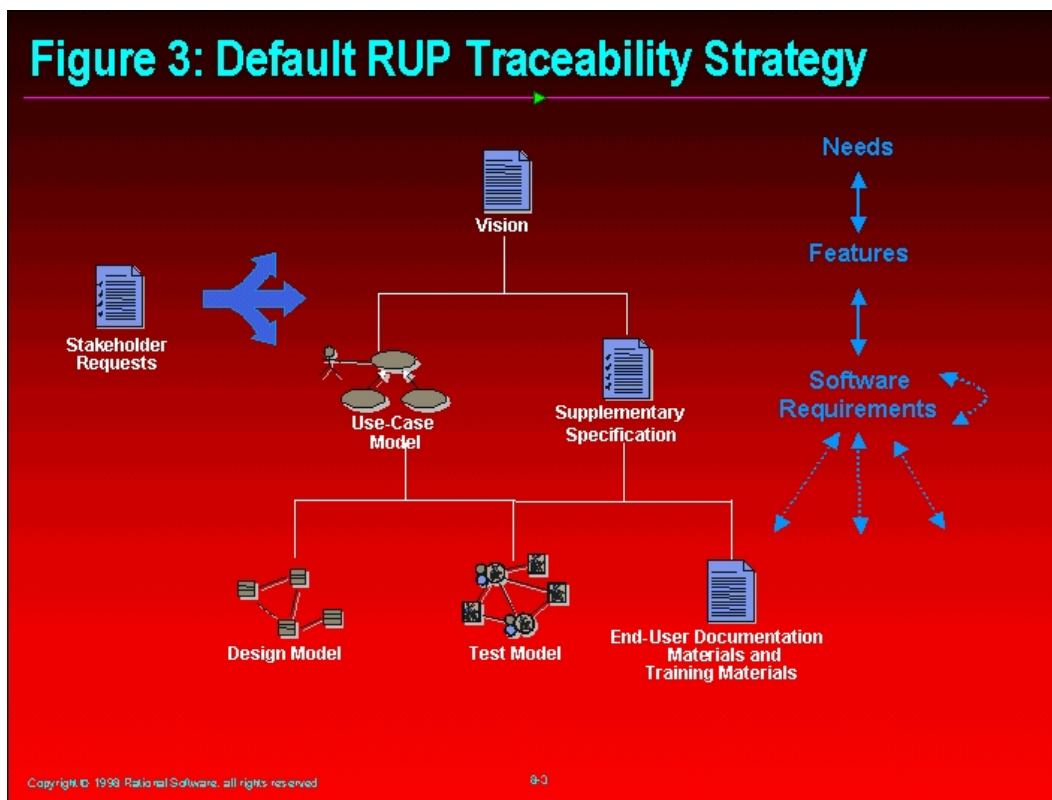
Esistono naturalmente molte altre opzioni tra le quali quella di non utilizzare affatto il modello di caso d'uso. Tale opzione viene chiamata "Nessun modello di caso d'uso".

Questi ed altri approcci vengono affrontati in dettaglio nel catalogo delle strategie di tracciabilità seguente.

### Perché adottare approcci ibridi?

Queste due soluzioni estreme (Modello di caso d'uso unico e Nessun modello di caso d'uso) acquisiscono un punto di vista davvero puro solo consentendo una forma di cattura dei requisiti. Entrambe presumono che un approccio unico sia applicabile per tutti i progetti e tutte le comunità di stakeholder. Entrambe hanno condiviso successi ma hanno fallito a causa della loro incapacità e mancanza di flessibilità a gestire tutte le situazioni complesse e l'insieme delle relazioni degli stakeholder che sorgono nei progetti "reali".

Rational Unified Process raccomanda la seguente gerarchia di tracciabilità.



Si tratta dell'opzione "Caratteristiche che guidano il modello di caso d'uso" ed è probabilmente la strategia di tracciabilità più efficace, ma bisogna notare che anche nel caso dell'adozione di Rational Unified Process non sempre è così.

Esempi di utilizzo della modellazione di caso d'uso come unico meccanismo per la specifica di requisiti software funzionali che è rivelato problematico sono:

- ☐ Il caso in cui ci sono troppe fonti di requisiti contraddittorie (ad esempio, molti desideri contraddittori dovranno essere controllati e gestiti).
- ☐ Se il progetto viene perseguito all'interno di un'organizzazione che insiste sulla conformità con un processo di cattura dei requisiti tradizionale esistente.
- ☐ Se si verificano problemi nel far essere presenti gli stakeholder nella modellazione dei workshop per la produzione di un singolo modello di consenso per i requisiti.
- ☐ Se la modellazione del caso d'uso deve essere utilizzata per abilitare lo sviluppo di software Object-Oriented all'interno dei vincoli di un processo di cattura dei requisiti esistente.
- ☐ Se la comunità degli stakeholder non è in grado o è poco propensa ad esprimere tutti i propri desideri di livello dei requisiti software funzionali direttamente all'interno del modello di caso d'uso.
- ☐ Se il cliente ha definito che il prodotto debba essere distribuito nel formato di una serie di requisiti software tradizionali. È una situazione molto comune quando lo sviluppo viene presentato per l'acquisto - la tradizionale istruzione di requisito diviene allora parte del contratto di distribuzione.

È opinione diffusa che la decisione sull'approccio da adottare va presa all'interno del contesto di ogni progetto e organizzazione di sviluppo. Non esiste un'unica soluzione al problema e sarebbe da pazzi cercare di forzare tutti i progetti in un unico approccio alla gestione dei requisiti.

Va ricordato che Rational Unified Process è un processo configurabile e può gestire tutte le strategie di tracciabilità presentate in questo documento tranne per l'approccio di "Nessun modello di caso d'uso" (la natura guidata del caso d'uso di



Rational Unified Process preclude l'adozione di quest'opzione). Quale approccio adottare è una delle decisioni da prendere durante la produzione del caso di sviluppo di Rational Unified Process.

## Informazioni relative al catalogo di strategia della tracciabilità

---

Per poter definire strategie di tracciabilità è necessario un meccanismo per le categorie e la definizione degli elementi di tracciabilità:

**Definizione: tipo di tracciabilità**

Una classificazione in categorie del tempo di tracciabilità (ad esempio necessità, funzioni del prodotto, casi d'uso, requisiti software, requisiti di test, attori, termini di glossario e così via) in base a caratteristiche ed attributi comuni.

**Nota:** In RequisitePro, i tipi di tracciabilità sono rappresentati dai tipi di requisiti. Impostare

una strategia di tracciabilità coinvolge tre attività strettamente unite:

- ☐ identificare la serie di tipi di tracciabilità richiesti per definirne gli elementi.
- ☐ identificare le valide relazioni di tracciabilità tra questi tipi.
- ☐ identificare gli attributi richiesti da questi elementi per abilitare una gestione efficace dei requisiti per il progetto.

Il catalogo della strategia di tracciabilità facilita i primi due di questi passi documentando serie note di elementi di tracciabilità e le loro relazioni di tracciabilità. Non copre la terza attività poiché la definizione degli attributi appropriati per i tipi di tracciabilità si trova, attualmente, al di là dell'ambito del presente documento).

Le strategie di tracciabilità descritte nel catalogo fanno tutte uso di sottoinsiemi della stessa serie di base dei tipi di tracciabilità.

- ☐ Necessità
- ☐ Caratteristiche del prodotto
- ☐ requisiti software (funzionali e non) ☐ Elementi di glossario
- ☐ Casi d'uso
- ☐ Sezioni di caso d'uso
- ☐ Attori

**Nota:** spesso nella modellazione del caso d'uso gli unici requisiti software sono quelli supplementari definiti dalla specifica supplementare.

La possibilità di poter potenzialmente tracciare tra tutti i tipi di tracciabilità tradizionale e le parti di componente del modello di caso d'uso favorisce una gran quantità di strategie disponibili per il progetto.

Esistono due livelli di tracciabilità tra i relativi elementi:

- ☐ Tracciabilità di base

Esiste una tracciabilità di base applicabile qualunque sia la strategia scelta. È implicita nella natura dei tipi di tracciabilità. Copre cose come la relazione tra casi d'uso e sezioni di casi d'uso o tra casi d'uso e attori.

Nella lettura dei diagrammi generali nella sezione delle strategie di tracciabilità qui sotto questa tracciabilità di base non viene ripetuta per ogni strategia ma viene inclusa in tutte quelle applicabili come valore predefinito.

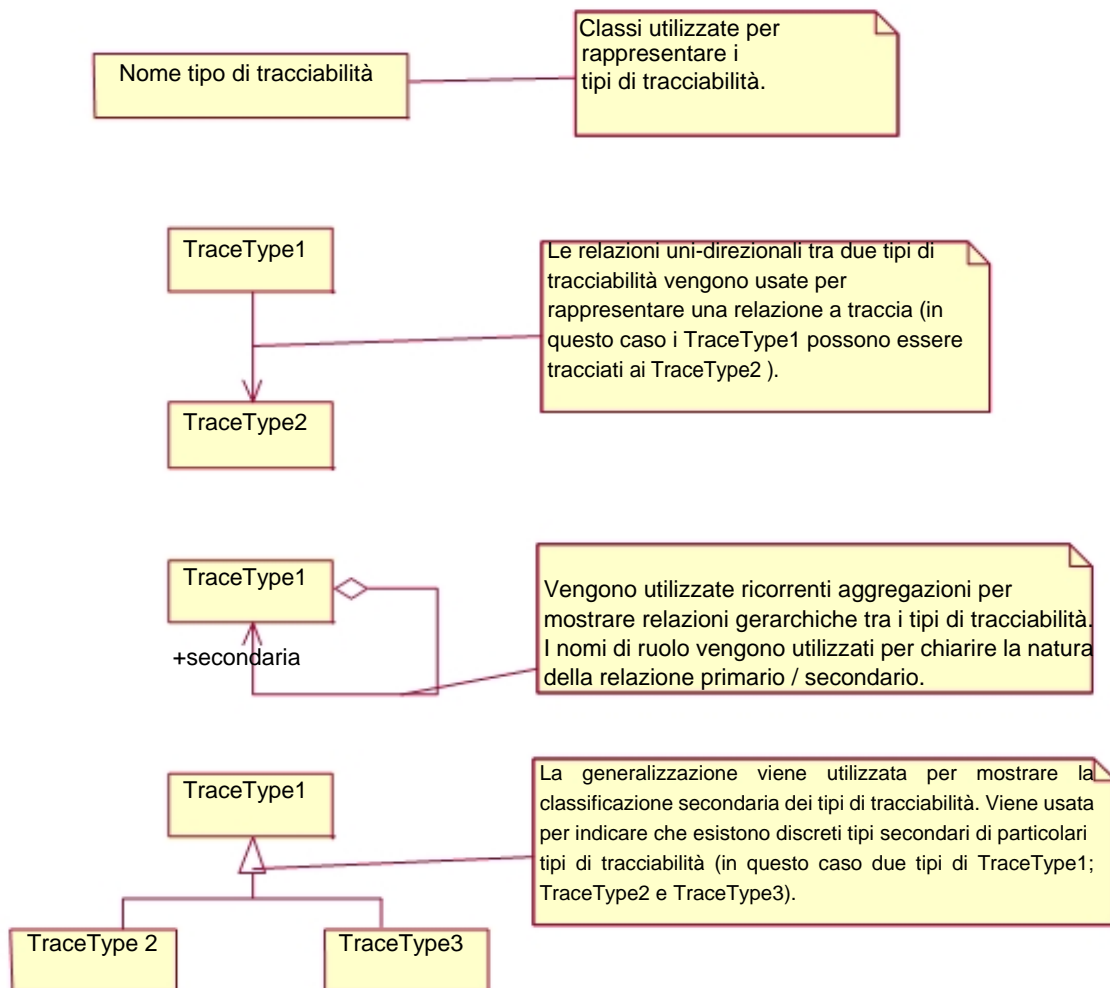
- ☐ Tracciabilità estesa

È la tracciabilità introdotta a supporto di una strategia specifica. È maggiormente soggettiva e varia con le varie strategie.

## Catalogo di strategia della tracciabilità

### Annotazione diagrammi

I tipi di tracciabilità e le loro relazioni sono mostrati come diagrammi UML (Unified Modeling Language). La figura sottostante illustra come interpretare l'uso di UML in questo contesto.



Per comprendere appieno i diagrammi è utile conoscere la mappatura usata durante l'implementazione delle definizioni in RequisitePro. La tabella sotto riportata spiega in che modo l'annotazione di diagramma possa essere associata ai progetti RequisitePro.

Annotazione diagrammi	Mappatura di RequisitePro
Tipi di classe / tracciabilità	Tipo di requisito
Relazioni	Relazioni di "traccia" di RequisitePro
Aggregazioni	Requisiti gerarchici
Generalizzazione	Classificazione del tipo di requisito super mediante aggiunta di attributo "di classificazione secondaria".

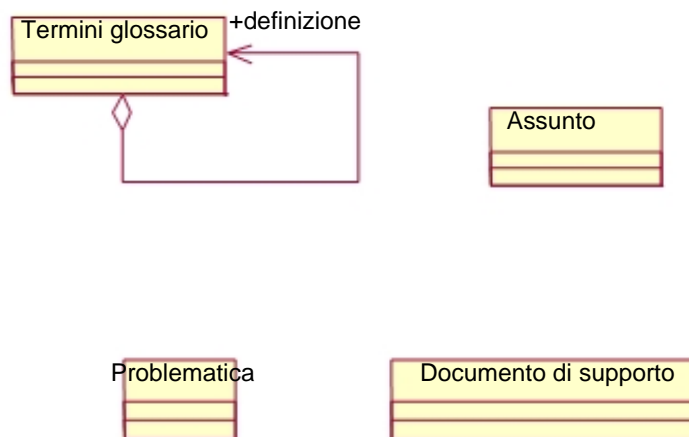
Nota: RequisitePro consente ad ogni elemento di tracciabilità di essere tracciato ad ogni altro elemento. Ciò che la strategia di tracciabilità definisce sono i collegamenti significativi che si trovano al centro della strategia di gestione dei requisiti del progetto.

#### Tipi di tracciabilità di supporto

##### Descrizione

In questa sezione viene definita una serie di tipi di tracciabilità di supporto che possono essere utilizzati per ogni strategia selezionata.

##### Panoramica



## Tipi di tracciabilità

Tipo di tracciabilità	Descrizione
Termini di glossario	Questo tipo di tracciabilità definisce i relativi elementi che rappresentano i termini di glossario e le loro definizioni.  viene incluso nella serie di tipi di tracciabilità di supporto, come un glossario viene richiesto indipendentemente dalla strategia che si sceglie di adottare.
Problematica	Questo tipo consente di aggiungere elementi di tracciabilità che rappresentano le problematiche da tracciare all'interno di RequisitePro. Queste problematiche possono poi essere associate a tutti gli elementi di tracciabilità su cui hanno effetto.  Un esempio di utilizzo del tipo di tracciabilità della problematica può essere la traccia di problematiche associate agli elementi del glossario. Se una definizione non è certa o se è in discussione, le problematiche possono essere evidenziate ed incluse in RequisitePro. Questo assicura che la questione non venga dimenticata e consente una vista ai report costruiti su tutti gli elementi di glossario con problematiche eccezionali. Un altro buon utilizzo di questo tipo di tracciabilità consiste nel tracciare questioni sorte durante la revisione dei casi d'uso ed altri artefatti di sviluppo.
Assunto	Questo tipo di tracciabilità consente la traccia di assunti fatti. Tali assunti possono poi essere associati con qualsiasi elemento di tracciabilità che hanno influenzato.
Documento di supporto	Questo tipo di tracciabilità consente di aggiungere tutti i documenti preferiti nella gerarchia di tracciabilità. È particolarmente utile per l'inclusione di esempi pre-esistenti o documentazione che chiarisce il significato o lo scopo di un altro elemento di tracciabilità. I meccanismi di tracciabilità flessibile di RequisitePro consente l'associazione di documentazione di supporto con ogni tipo di elemento di tracciabilità.  Un esempio di utilizzo del tipo di documento di supporto consiste nell'includere specifiche di messaggio EDI dettagliate come informazioni di supporto per il glossario oppure come appendici ai casi d'uso destinati all'utilizzo dei messaggi.

## Tracciabilità di base

Collegamento alla tracciabilità	Descrizione
Da termini di glossario a termini di glossario	La relazione consente di catturare sia il nome del termine di glossario che la sua definizione mediante un singolo tipo di tracciabilità.
Tipo di tracciabilità di supporto per ogni altro tipo di tracciabilità	Questi tipi di tracciabilità di supporto possono essere tracciati ad uno qualsiasi degli altri tipi coinvolti nella strategia scelta.

## Nessun modello di caso d'uso

### Descrizione

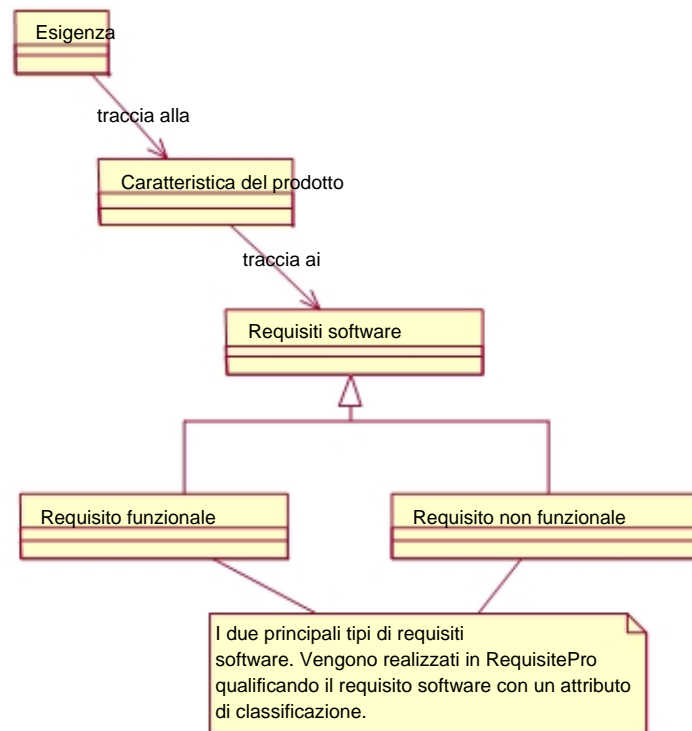
In questo caso manca il modello di caso d'uso. Le necessità danno origine alle funzioni del prodotto, che a turno danno origine ai requisiti software documentati in un SRS (Software Requirements Specification) formale.

Tipizzato da responsabili di progetto che dicono, "Non ho bisogno di un modello di caso d'uso che puzza!"

### Caratteristiche

Caratteristica	Valore	Commento
Tracciabilità esplicita	Alto	I progetti che praticano le tecniche di gestione dei requisiti senza i casi d'uso mantengono in genere un alto livello di tracciabilità esplicita tra i relativi tipi.
Fiducia	Basso	
Responsabilità	Alto	
Formalità	Alto	
Completezza	Basso	È molto difficile valutare una serie di requisiti software tradizionali per la completezza.
Serie di documenti	Grande	La serie di documenti viene di solito misurata in piedi e non in pollici.
Focalizzazione Contratto		La focalizzazione del processo di cattura dei requisiti è sullo stabilire un contratto legalmente attuabile tra i clienti e gli sviluppatori piuttosto che sullo stabilire una comprensione condivisa del problema da risolvere e la soluzione proposta.
Comprensibilità	Basso	La documentazione dei requisiti è spesso inaccessibile sia alla comunità utenti che agli sviluppatori. È costituito in genere da molte voci di riga individuali raggruppate per tipo o per area funzionale che forniscono poco contesto per i revisori.
Processo	Tipicamente Cascata	Le tecniche di cattura dei requisiti tradizionali sono spesso messe in pratica come parte di un processo di sviluppo a cascata. La mancanza di contesto per e la difficoltà di valutazione della completezza di tutti i sottoinsiemi dei requisiti non facilita l'adozione di processi di sviluppo iterativi ed incrementali.
Stile di sviluppo	Funzionale Decomposizione	Il raggruppamento dei requisiti per tipo o area funzionale tende ad una decomposizione funzionale continua quando i requisiti vengono trasferiti in una soluzione.

## Panoramica di tracciabilità



## Tipi di tracciabilità

Tipo di requisito	Descrizione
Esigenza	Il problema operativo o di business (opportunità) da risolvere per giustificare l'acquisto o l'uso. Anche noto come obiettivo o risultato.
Caratteristica prodotto	Una capacità o caratteristica di un sistema che direttamente soddisfa un'esigenza. Spesso ritenuti "benefici promossi" del sistema.
Requisiti software	Una condizione o capacità alla quale il software in costruzione deve conformarsi.

## Riepilogo tracciabilità

Collegamento alla tracciabilità	Descrizione
Le esigenze tracciano verso le funzioni del prodotto	Ogni esigenza viene realizzata da una serie di caratteristiche. Questa relazione consente il vantaggio business di ogni caratteristica da tracciare.
Le funzioni del prodotto tracciano verso i Requisiti software.	Ogni caratteristica viene realizzata da una serie di requisiti software. Questa relazione consente la traccia del vantaggio business di ogni requisito software ed abilita la gestione dell'ambito dei requisiti software al livello della funzione del prodotto.

## Vantaggi e svantaggi

### Pro:

- ☐ Ben compreso
- ☐ È ritenuto buono per i contratti legali (vedere i continui casi di tribunale relativi alla capacità o meno del software consegnato di soddisfare i requisiti specificati).
- ☐ Raccomandato da molti processi standard.
- ☐ Abilita tracciabilità dettagliata, rigorosa e di basso livello.
- ☐ Non altera lo status quo introducendo proposte "inutili" ideas

### Contro:

- ☐ Difficile completare la cattura dei requisiti - è molto facile restare bloccati nella fase dei requisiti. ☐ Difficile comprendere i requisiti espressi in questo formato.
- ☐ L'analisi d'impatto della modifica dei requisiti è difficile da valutare.
- ☐ I requisiti individuali non hanno contesto.
- ☐ Alti costi di mantenimento. La mancanza di tracciabilità implicita lascia ai progetti i costi per il mantenimento di grandi quantità di relazioni di tracciabilità esplicita.
- ☐ La mancanza di contesto rende difficile identificare significativi sottoinsiemi dei requisiti. Tutto questo rende, a turno più difficile la gestione dell'ambito e la distribuzione incrementale del prodotto.

## Esempi

L'approccio al modello senza caso d'uso per la tracciabilità dei requisiti viene ampiamente usato in diversi progetti di molte aree business. Molte organizzazioni richiedono una SRS tradizionale e rigorosa come base di una negoziazione contrattuale formale. Questo le induce a pensare che l'approccio alla gestione dei requisiti tradizionale sia l'unico appropriato per i propri progetti.

## Modello di caso d'uso unico

### Descrizione

"Il modello di caso d'uso rappresenta i miei requisiti." Una stretta associazione ed un alto livello di fiducia tra i clienti e gli sviluppatori caratterizzano in genere i progetti che adottano tale approccio. Viene normalmente usato per progetti interni e di bassa responsabilità in cui gli sviluppatori sperano di dimostrare o ottenere una chiara comprensione dei requisiti sviluppando il modello di caso d'uso insieme ai (o approvato dai) clienti.

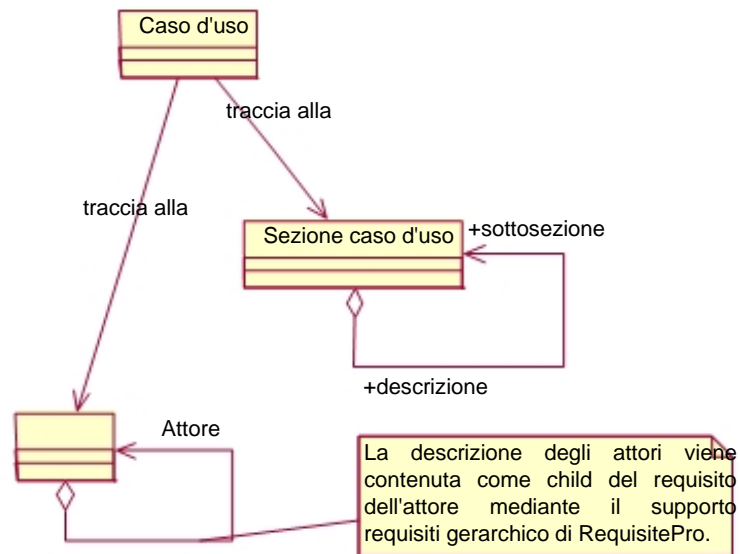
In questo caso il modello di caso d'uso è l'unica istruzione dei requisiti dei sistemi. Il modello di caso d'uso, il glossario e le specifiche supplementari formano l'intera istruzione dei requisiti di sistema.

## Caratteristiche

Caratteristica	Valore	Commento
Tracciabilità esplicita	Basso	Nessuna tracciabilità esplicita richiesta. La tracciabilità implicita fornita adottando un approccio guidato al caso d'uso viene considerata sufficiente. Probabilmente non viene mantenuta alcuna tracciabilità esplicita, perciò non viene utilizzato alcun tool di gestione dei requisiti.
Fiducia	Alto	La mancanza di esigenze o di analisi del livello delle caratteristiche significa che agli sviluppatori del modello di caso d'uso viene concessa grande fiducia dagli stakeholder per la produzione del giusto software.
Responsabilità	Basso	
Formalità	Basso	
Completezza	Basso	Anche se uno stesso modello di caso d'uso facilita lo stabilire completezza di SRS (Software Requirements Specification) la mancanza di tracciabilità alle esigenze dello stakeholder spesso conduce alla produzione di un sistema incompleto o elaborato.
Serie di documenti	Piccolo	Questo approccio coinvolge la serie di documenti minima II.
Focalizzazione	Utente	modello di caso d'uso ha una prospettiva utente
Comprensibilità	Alto	È facile da comprendere per tutti gli stakeholder nel progetto.
Processo	Tipicamente Iterativo e Incrementale	I casi d'uso inseriscono i requisiti software nel contesto favorendo lo sviluppo iterativo e incrementale (il caso d'uso fornisce una buona unità di distribuzione). I casi d'uso possono anche essere utilizzati con processi di sviluppo a cascata.
Stile di sviluppo	Tipicamente Object Oriented	In genere vengono utilizzati per guidare sviluppi software Object-Oriented anche se funzionano in tutti i casi. Se uno stile OO non viene adottato allora viene richiesto un alto livello di tracciabilità esplicita.



## Panoramica di tracciabilità



## Tipi di tracciabilità

Tipo di tracciabilità	Descrizione
Caso d'uso	Questo tipo di tracciabilità definisce i relativi elementi dei casi d'uso
Sezione del caso d'uso	<p>La sezione del caso d'uso abilita l'inclusione delle sezioni dei casi d'uso nella gerarchia di tracciabilità.</p> <p>Questo consente la traccia all'interno di flussi individuali ed altre proprietà che includono il caso d'uso.</p> <p>La presenza di relazione gerarchica alle sezioni secondarie consente la cattura di parti singole di ogni sezione. Questo ad esempio consente l'identificazione delle pre-condizioni che creano la sezione delle condizioni precedenti.</p> <p>Nota: in alcuni casi è possibile applicare l'identificazione di singoli requisiti software all'interno del flusso di eventi del caso d'uso (in questo caso sezioni molto piccole) ma non è consigliabile finché lo stesso caso d'uso non è stabile.</p>
Attore	<p>Questo tipo di tracciabilità definisce i relativi elementi che rappresentano Gli attori.</p>

## Riepilogo di tracciabilità

Collegamento alla tracciabilità	Descrizione
Dal caso d'uso alla sezione di caso d'uso	Ogni caso d'uso è costituito da una serie di sezioni del caso d'uso. Questa relazione consente di rintracciare sezioni di caso d'uso che creano uno specifico caso d'uso.
Dalla sezione di caso d'uso al caso d'uso Sezione	Alcune delle sezioni più complesse del caso d'uso sono costituite da molte sezioni secondarie. Ad esempio un flusso di eventi può essere costituito da molti flussi secondari, così come la sezione relativa alle condizioni precedenti da molte pre condizioni.
Dal caso d'uso all'attore	Questa relazione consente di vedere gli attori coinvolti un uno specifico caso d'uso.
Da attore ad attore	Consente anche la cattura del nome dell'attore e di una sua breve descrizione mediante un singolo tipo di tracciabilità.

## Vantaggi e svantaggi

### Pro:

- ☐ Serie di documenti minima
- ☐ Minimo impegno nella gestione dei requisiti
- ☐ Buon supporto alla gestione dell'ambito, l'analisi dell'impatto e lo sviluppo incrementale. ☐ I casi d'uso sono facili da capire.

### Contro:

- ☐ Non c'è relazione relativa alle esigenze dello stakeholder. Non viene effettuato alcun serio tentativo di analisi del problema prima di iniziare la definizione della soluzione.
- ☐ Per alcuni è difficile accettare un contratto basato solo su un modello di caso d'uso.
- ☐ Senza intraprendere analisi sulle esigenze può essere difficile capire se il modello di caso d'uso descrive una soluzione idonea. È facile distrarsi durante la scrittura di casi d'uso.
- ☐ Nell'esecuzione di rilasci regolari può diventare difficile gestire continuamente le aspettative degli stakeholder senza informazioni ad un livello superiore ai casi d'uso.

## Esempi

Questo approccio viene in genere utilizzato per progetti di piccola scala, informali e interni, in cui sviluppatori e utenti lavorano a stretto contatto.

## Il modello di caso d'uso definisce le funzioni del prodotto

### Descrizione

In questo caso la modellazione del caso d'uso viene utilizzata come principale metodo di deduzione dei requisiti ed il modello di caso d'uso diventa la definizione delle funzioni del prodotto che il sistema deve fornire così come l'istruzione dei requisiti software.

Questa opzione è adatta solo per piccoli sviluppi, con brevi cicli di vita e senza altra scala. Se anche ogni caso d'uso rappresentasse una caratteristica del sistema le prime sarebbero più delle seconde e, in realtà, molte possono avere effetto su

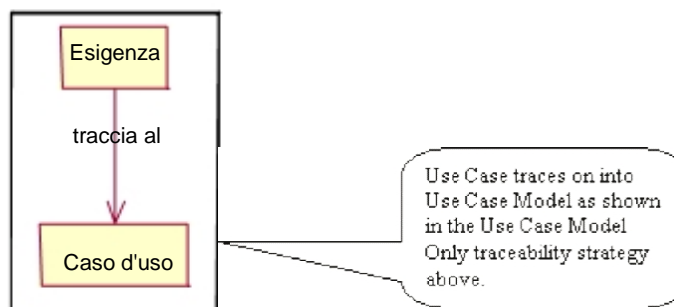
diversi casi d'uso. Con l'evolversi del sistema è sempre meno probabile che le nuove caratteristiche prendano la forma di nuovi casi d'uso.

### Caratteristiche

È una variazione del precedente approccio "Modello di caso d'uso unico". Sono state rilevate solo poche differenze durante la discussione su questo approccio.

Caratteristica	Valore	Commento
Tracciabilità esplicita	Basso	Come l'approccio del modello di caso d'uso unico. Esiste solo una piccola quantità di esigenze e questa serie di tracciabilità aggiuntiva è costituita da piccole quantità di tracciabilità esplicita recuperata.
Fiducia	Medio / Alto	L'aggiunta di esigenze al modello di caso d'uso ha come risultato una strategia leggermente meno affidabile rispetto al modello di caso d'uso Unico
Responsabilità	Basso	
Formalità	Basso	
Completezza	Medio	Il modello di caso d'uso stesso favorisce la definizione della completezza di SRS (Software Requirements Specification), la tracciabilità aggiuntiva rispetto alle esigenze dello stakeholder favoriscono la convalida dell'applicabilità del modello di caso d'uso.
Serie di documenti	Piccolo	Questo approccio coinvolge una serie di documenti minima. Modello di caso d'uso più il documento di Visione contenente le esigenze.
Focalizzazione	Utente	Il modello di caso d'uso e le esigenze hanno entrambe una prospettiva utente.
Comprensibilità	Alto	Le esigenze ed il modello sono semplici da comprendere per tutti gli stakeholder nel progetto.
Processo Stile sviluppo iterativo e incrementale	Tipicamente	Come l'approccio del modello di caso d'uso unico
	Tipicamente Object Oriented	Come l'approccio del modello di caso d'uso unico

### Panoramica di tracciabilità



### Tipi di tracciabilità

Tipo di tracciabilità	Descrizione
Esigenza	Come definito per l'approccio "nessun modello di caso d'uso"
Caso d'uso	Come definito per l'approccio "modello di caso d'uso unico"

### Riepilogo di tracciabilità

Collegamento alla tracciabilità	Descrizione
Dall'esigenza al caso d'uso	In questo caso le esigenze tracciano direttamente ai casi d'uso. L'assunto è che i casi d'uso possano svolgere il ruolo di funzioni del prodotto durante la sua gestione e quella dell'ambito.

### Vantaggi e svantaggi

Questo approccio è molto simile alla strategia "modello di caso d'uso unico". Vantaggi e svantaggi sono gli stessi con le seguenti aggiunte e rischi.

#### Pro:

- ☐ In questo caso il modello di caso d'uso fa riferimento alle esigenze dello stakeholder che favoriscono la valutazione della idoneità del modello.

#### Contro:

- ☐ I casi d'uso possono apparire per definire le caratteristiche del sistema nelle prime fasi del progetto ma i due concetti divergeranno con la maturazione dello stesso.
- ☐ I casi d'uso non sono caratteristiche - ciò che sembra una strategia per risparmiare tempo e lavoro diventa rapidamente una ingestibile confusione.

### Esempi

Anche se sono stati effettuati tentativi di utilizzo di questa strategia di tracciabilità su piccoli progetti interni, tale approccio non è consigliato a causa della scalabilità e dei problemi di evoluzione del prodotto nel lungo termine. È consigliata l'adozione della strategia "Caratteristiche che guidano il modello di caso d'uso" se il modello va integrato con la tracciabilità risalente le esigenze dello stakeholder.

### Le caratteristiche guidano il modello di caso d'uso

#### Descrizione

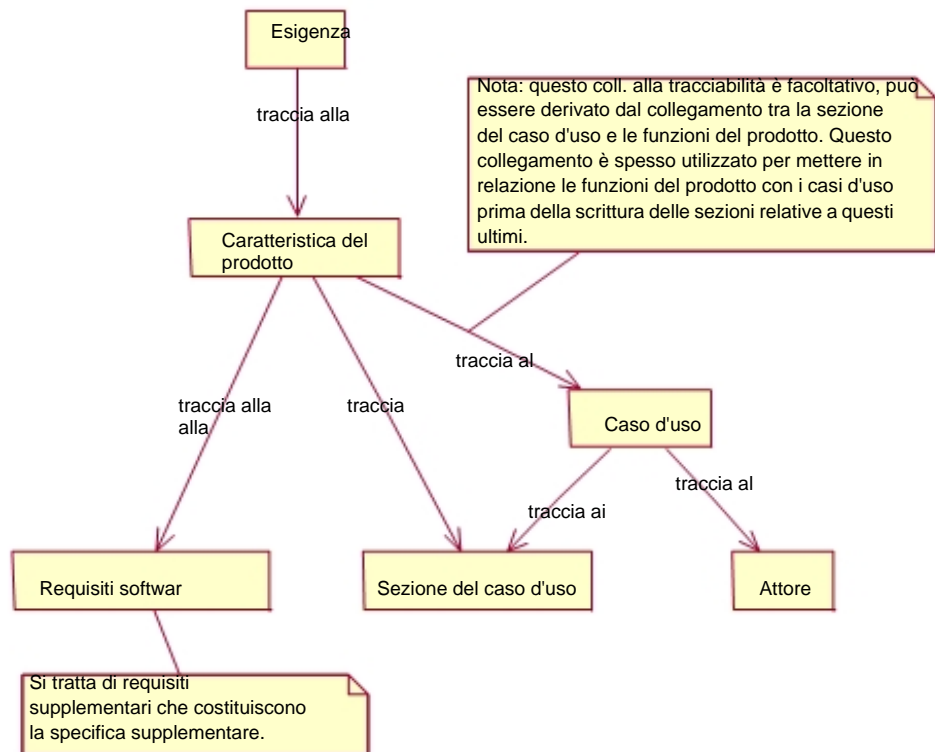
Il modello di caso d'uso e le specifiche supplementari comprendono SRS." È la strategia consigliata e delineata da RUP. Esigenze e funzioni del prodotto vengono documentate nel documento di Visione e tracciate verso i casi d'uso. Se non sono riflesse nel modello di caso d'uso, vengono tracciate nelle specifiche supplementari.

In questo caso il modello di caso d'uso agisce da istruzione principale dei requisiti software. Viene completato da una specifica supplementare contenente i requisiti software che non possono essere espressi facilmente nei casi d'uso stessi.

## Categorie

Caratteristica	Valore	Commento
Tracciabilità esplicita	Medio	Oltre alla tracciabilità implicita del modello di caso d'uso, in questo caso è necessario mantenere esplicitamente la tracciabilità tra le esigenze, le caratteristiche ed il modello di caso d'uso.
Fiducia	Medio	
Responsabilità	Alto	
Formalità	Medio	L'aggiunta di esigenze e funzioni del prodotto al modello di caso d'uso conduce ad un processo di gestione dei requisiti più formale del semplice mantenimento di un modello di caso d'uso.
Completezza	Alto	Avere prospettiva di caso d'uso e delle caratteristiche sui requisiti software rende possibile conseguire un alto livello di completezza rispetto la cattura e la classificazione per priorità dei requisiti stessi.  Adesso esiste un documento di Visione contenente le esigenze e le
Serie di documenti	Medio	caratteristiche, un modello di caso d'uso ed una specifica supplementare.  L'aggiunta di esigenze e caratteristiche al modello di caso d'uso amplia la
Focalizzazione	Utenti, Stakeholder e progetto Responsabili	focalizzazione dell'attività dei requisiti per inglobare più attivamente il responsabile di prodotto e tutti gli altri stakeholder, nonché gli utenti. Le caratteristiche rappresentano un tool molto potente per la gestione delle aspettative degli stakeholder e forniscono un buon complemento alla prospettiva di caso d'uso dei requisiti software.
Comprensibilità	Alto	La definizione di esigenze e caratteristiche accanto ad un modello di caso d'uso con specifiche supplementari fornisce modello di requisiti facilmente comprensibile da tutti gli stakeholder nel progetto.
Processo	Tipicamente Iterativo e Incrementale	Come l'approccio del modello di caso d'uso unico.
Stile di sviluppo	Tipicamente Object Oriented	Come l'approccio del modello di caso d'uso unico.

## Panoramica di tracciabilità



## Tipi di tracciabilità

Tipo di tracciabilità	Descrizione
Esigenza	Come definito per l'approccio "nessun modello di caso d'uso"
Caratteristica del prodotto	Come definito per l'approccio "nessun modello di caso d'uso"
Caso d'uso	Come definito per l'approccio "modello di caso d'uso unico"
Requisiti software	Tutti i requisiti software che vengono applicati all'intero sistema o che non si adattano facilmente in un caso d'uso. Se un requisito software è una condizione o una capacità a cui il software in costruzione deve conformarsi.
Sezione del caso d'uso	Come definito per l'approccio "modello di caso d'uso unico"
Attore	Come definito per l'approccio "modello di caso d'uso unico"

## Riepilogo di tracciabilità

Collegamento alla tracciabilità	Descrizione
Dall'esigenza alla funzione del prodotto	Come definito per l'approccio "nessun modello di caso d'uso"
Dalla funzione del prodotto al caso d'uso	Collegamento alla tracciabilità facoltativo. La funzione del prodotto può tracciare direttamente ai casi d'uso. Questo consente alle funzioni del prodotto di essere assegnate ai casi d'uso prima della scrittura delle relative sezioni e di eseguire analisi d'impatto sul modello di caso d'uso al livello della funzione del prodotto e viceversa.
Dalla funzione del prodotto alla sezione di caso d'uso*	Le funzioni del prodotto tracciano alle sezioni del caso d'uso. Questo consente al modello di caso d'uso di essere gestito su una base di funzioni e favorisce l'analisi dell'impatto tra la serie di funzioni ed il modello di caso d'uso ad un livello più applicabile del caso d'uso stesso.  Tutte le funzioni del prodotto tracciate verso un caso d'uso devono anche tracciare verso una delle sezioni del caso d'uso.
Dalla funzione del prodotto ai requisiti software*	Le funzioni del prodotto tracciano anche ai requisiti software nella specifica supplementare. Tutte le funzioni del prodotto che non tracciano in un modello di caso d'uso devono tracciare ad almeno uno dei requisiti software nella specifica supplementare.
Dal caso d'uso all'attore	Come definito per l'approccio del modello di caso d'uso unico.
Da caso d'uso alla sezione di caso d'uso	Come definito per l'approccio del modello di caso d'uso unico.

\*Ogni funzione del prodotto deve tracciare ad almeno una sezione di caso d'uso o SRS (Supplementary Software Requirement). Se questo non avviene non viene inclusa nel software.

## Vantaggi e svantaggi

Questo approccio massimizza i vantaggi forniti dagli approcci di gestione dei requisiti tradizionale e di caso d'uso minimizzandone gli svantaggi.

Pro:

- ☐ Ben compreso
- ☐ Raccomandato da Rational Unified Process.
- ☐ Abilita tracciabilità dettagliata, rigorosa e di basso livello.
- ☐ Avere sia una prospettiva di caso d'uso che di funzione del prodotto sui requisiti software favorisce il completamento della cattura dei requisiti - questo diminuisce le possibilità di arresto durante le attività di deduzione e cattura dei requisiti.
- ☐ I requisiti software vengono espressi in un formato facile da comprendere.
- ☐ L'analisi d'impatto della modifica dei requisiti è favorita da questa strategia di tracciabilità - l'impatto nel non implementare una funzione o una sezione di caso d'uso può essere chiaramente compresa.
- ☐ I requisiti individuali hanno un contesto fornito dai casi d'uso e/o dalle funzioni del prodotto. Questo rende facile l'identificazione di significativi sottoinsiemi dei requisiti. Ciò a turno rende la gestione dell'ambito e la consegna incrementale del prodotto molto più semplice.
- ☐ Serie di documenti completa minima.
- ☐ Diminuisce l'impegno speso nella gestione dei requisiti.

- ☐ La soluzione scala bene. Se vengono eseguiti regolari rilasci la capacità di gestione dell'ambito sia al livello di funzione che a quello di caso d'uso consente a tutti gli stakeholder di seguire il progresso del progetto al livello di dettaglio ritenuto appropriato.
- ☐ In questo caso il modello di caso d'uso risale alle esigenze degli stakeholder attraverso le funzioni di prodotto che aiutano tutti loro a valutare l'idoneità del modello stesso.

Contro:

- ☐ Non accettabile da tutte le organizzazioni
- ☐ Per alcuni è difficile scrivere un contratto basato sui requisiti software espressi in gran parte come modello di caso d'uso anche se molte organizzazioni lo hanno conseguito con esito positivo.

### Esempi

Questo approccio è applicabile su tutti i progetti in cui i casi d'uso sono accettati come formato idoneo all'espressione della maggior parte dei requisiti software.

Il modello di caso d'uso è un'interpretazione di SRS (Software Requirements Specification)

### Descrizione

"Il modello di caso d'uso è un'interpretazione del formale SRS". Viene più spesso usato se questo SRS formale è un mandato di protocollo interno o regolatore.

Un SRS tradizionale viene spesso considerato una parte essenziale per il conseguimento di un accordo contrattuale se viene intrapreso o delocalizzato uno sviluppo a prezzo fisso. Questo conduce a due situazioni tipiche:

All'organizzazione di sviluppo viene fornito un SRS tradizionale, da parte del cliente, all'inizio dello sviluppo del sistema.

Il documento SRS è un componente distribuibile obbligatorio o normativo nella fase iniziale del ciclo di vita del progetto. Ogni progetto deve avere un documento SRS tradizionale, formale, espressione dei requisiti dei sistemi così come per tutti gli altri progetti.

In questi casi il modello di caso d'uso viene usato per modellare e re-interpretare tutti i requisiti software all'interno dell'ambito del progetto. Se questo approccio viene adottato è norma per SRS arrivare per primo - esistono altre tecniche disponibili per trasformare le informazioni contenute da un modello di caso d'uso in un formato che assomiglia ad un SRS tradizionale e formale (specie se è stato adottato l'approccio "Le funzioni guidano il modello di caso d'uso" ) senza creare una seconda definizione dei requisiti software.

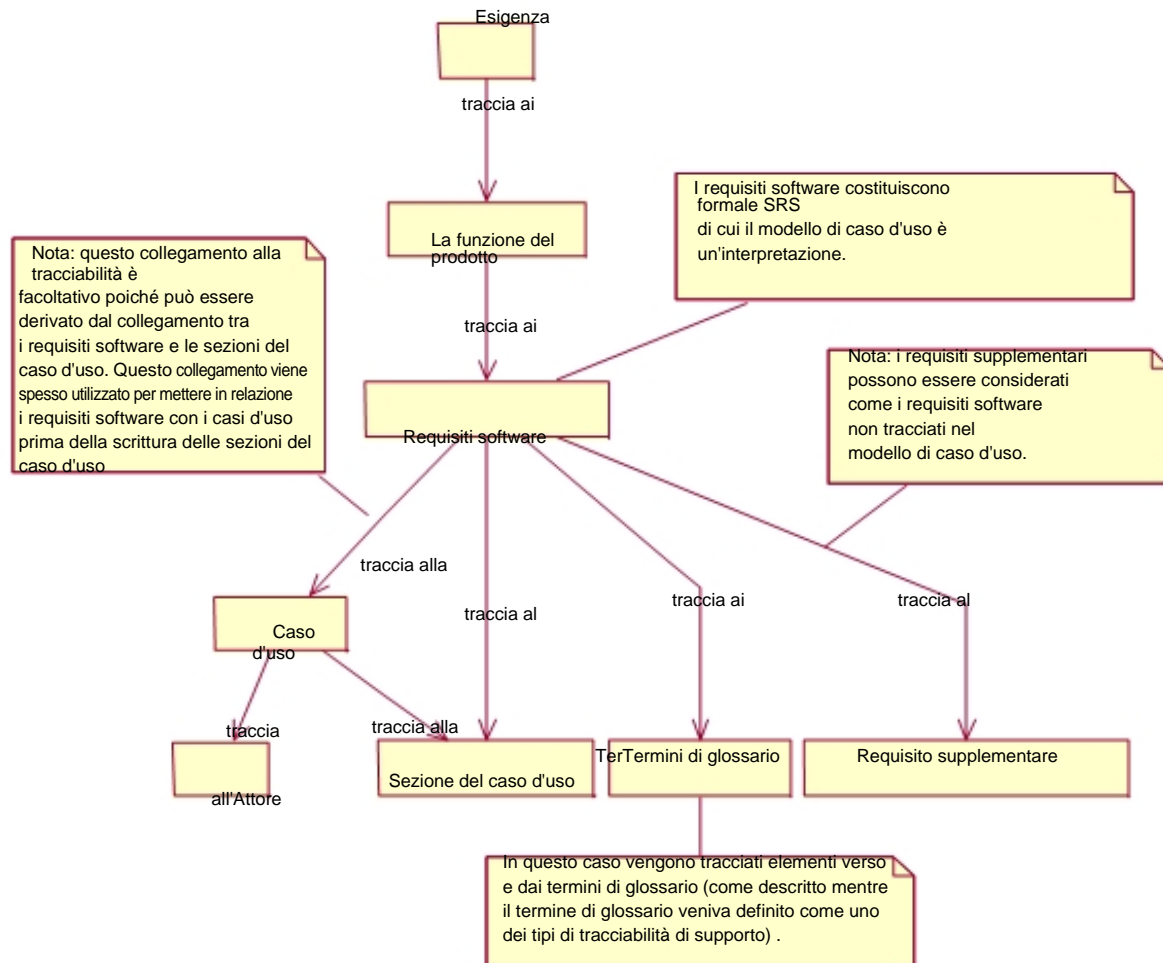
Nota: con tale approccio non è necessario che la serie di requisiti software "tradizionali" sia una istruzione completa della funzionalità richiesta - il modello di caso d'uso fornisce o assicura la completezza della specifica funzionale. I requisiti software "tradizionali" possono essere usati soltanto per la cattura di quelli direttamente identificati o evidenziati dagli stakeholder.



## Caratteristiche

Tracciabilità esplicita	Molto Alto	Tutta la tracciabilità richiesta dall'approccio "Nessun modello di caso d'uso" è necessaria per mantenere l'SRS in più esiste la spesa aggiuntiva nel tracciare i requisiti software tradizionali nel modello di caso d'uso.
Fiducia	Molto Bassa	Questo è un approccio molto "sicuro" che implica un livello di fiducia molto basso.
Responsabilità	Alto	
Formalità	Molto Alto	Di nuovo, questo è un approccio molto rigoroso costituito da due Strategie di gestione requisiti applicate in parallelo.
Completezza		Molto Alto Il complemento del tradizionale SRS con un modello di caso d'uso lo rende un approccio molto completo. Nota: in questo caso si assume che sia il modello di caso d'uso, che assicura una specifica completa della funzionalità del sistema. La serie di SRS (Software Requirement Specifications) non deve essere allo stesso livello di completezza.
Serie di documenti	Molto grande	In questa situazione il sistema viene fondamentalmente definito due volte.
Focalizzazione	Contratti	L'adozione di questo approccio viene guidato dall'esigenza di soddisfare un contratto esistente, espresso da un SRS tradizionale o di adattarsi ad una metodologia di sviluppo esistente che richiede un SRS come il contratto tra gli sviluppatori ed i clienti.
Comprensibilità	Media	La produzione di due definizioni di requisito software può essere confusa all'inizio ma l'uso del modello di caso d'uso come definizione principale dovrebbe condurre ad una istruzione facilmente comprensibile dei requisiti dei sistemi.
Processo	OPEN	Esiste abbastanza materiale a supporto di quasi ogni processo di sviluppo anche se il modello di caso d'uso viene spesso aggiunto ad un tradizionale SRS per abilitare l'uso di tecniche iterative e incrementali.
Stile di sviluppo	OPEN	Esiste abbastanza materiale a supporto di quasi ogni stile di sviluppo anche se il modello di caso d'uso viene spesso aggiunto ad un tradizionale SRS per abilitare l'uso di tecniche Object Oriented.

## Panoramica di tracciabilità



## Tipi di tracciabilità

Tipo di tracciabilità	Descrizione
Esigenza	Come definito per l'approccio "nessun modello di caso d'uso"
La funzione del prodotto	Come definito per l'approccio "nessun modello di caso d'uso"
Requisiti software	Come definito per l'approccio "nessun modello di caso d'uso"
Caso d'uso	Come definito per l'approccio "modello di caso d'uso unico"
Attore	Come definito per l'approccio "modello di caso d'uso unico"
Sezione del caso d'uso	Come definito per l'approccio "modello di caso d'uso unico"
Termini di glossario	Come definito per l'approccio "modello di caso d'uso unico"
Requisito supplementare	Tutti i requisiti software applicati all'intero sistema o che non si adattano facilmente in un caso d'uso. Possono non essere ripetute dalla serie originale dei requisiti software ma potrebbero solo essere nei requisiti software dell'ambito non tracciati al modello di caso d'uso.

## Riepilogo di tracciabilità

Collegamento alla tracciabilità	Descrizione
Dall'esigenza alla funzione del prodotto	Come definito per l'approccio "nessun modello di caso d'uso"
Dalla funzione del prodotto al software	Come definito per l'approccio "nessun modello di caso d'uso"
Requisito  Dal requisito software al caso d'uso	<p>I requisiti software funzionali vengono tracciati al caso d'uso. Anche un sottoinsieme di requisiti software non funzionali tracciano ai casi d'uso.</p> <p>Questa relazione consente un alto livello di definizione di ambito e di valutazione del modello di caso d'uso in termini dei suoi requisiti e vantaggi business.</p> <p>Nota: tutti i requisiti funzionali presenti nell'ambito <b>devono</b> tracciare verso i casi d'uso oppure al glossario. Se non vengono riflessi in questo modo non vengono implementati.</p>
Dal requisito software alla sezione del caso d'uso	<p>I requisiti software tracciati verso un caso d'uso devono inoltre tracciare ad una delle sue sezioni di caso d'uso.</p> <p>Questa relazione consente la verifica di sezioni di caso d'uso del caso d'uso con riferimento ai requisiti posti su di esso. Tutti i requisiti software tracciati verso un caso d'uso devono essere soddisfatti da una delle sezioni al suo interno. La doppia tracciabilità consente questa verifica e l'allocazione dei requisiti software agli stessi casi d'uso prima di considerare le sezioni di caso d'uso richieste.</p> <p>È possibile che alcune sezioni non abbiano corrispondenti requisiti software.</p> <p>Nota: Tutti i requisiti funzionali che tracciano verso un caso d'uso <b>must</b> tracciano anche verso una delle sue sezioni. Se non vengono riflessi in questo modo non vengono implementati.</p>
Dal requisito software ai termini di glossario	<p>Requisiti funzionali e non possono tracciare ad elementi nel Glossario. Questo è vero in particolare per i "requisiti statici" che identificano attributi e relazioni delle entità coinvolte nel sistema. Se viene tracciato un termine di glossario da un requisito software allora tale termine deve essere usato in uno dei casi d'uso oppure è improbabile che venga portato avanti nel modello di progettazione.</p>
Dal caso d'uso all'attore	Come definito per l'approccio "modello di caso d'uso unico"
Dal caso d'uso alla sezione di caso d'uso	Come definito per l'approccio "modello di caso d'uso unico"
d'uso  Dal requisito software ai requisiti supplementari	<p>I requisiti supplementari possono ribadire i requisiti software che vengono applicati all'intero sistema o che non si adattano facilmente al modello di caso d'uso. Un approccio alternativo consiste nel fare riferimento a tutti i requisiti software dell'ambito non tracciati al modello di caso d'uso come requisiti supplementari - questo impedisce il loro riavvio.</p>

## Vantaggi e svantaggi

Questo approccio deve essere eccezionalmente affidabile. È davvero appropriato per i progetti presentati con un SRS tradizionale e che desiderano utilizzare la modellazione del caso d'uso per ottenere una comprensione dei requisiti forniti e per favorire un approccio guidato al caso d'uso.

**Pro:**

- ☐ Abilita tracciabilità molto dettagliata, rigorosa e di basso livello.
- ☐ I requisiti software vengono espressi in un formato facile da comprendere.
- ☐ L'analisi d'impatto della modifica dei requisiti è favorita da questa strategia di tracciabilità - l'impatto nel non implementare una funzione, un requisito software o una sezione di caso d'uso può essere chiaramente compresa.
- ☐ I requisiti individuali hanno un contesto fornito dai casi d'uso e/o dalle funzioni del prodotto. La presenza del caso d'uso rende semplice identificare sottoinsiemi significativi dei requisiti. Tutto ciò a turno rende la gestione dell'ambito e la distribuzione incrementale del prodotto più facile.
- ☐ In questo caso il modello di caso d'uso viene fatto risalire alle esigenze dello stakeholder attraverso i requisiti software e le funzioni del prodotto che aiutano tutti gli stakeholder a valutare l'idoneità del modello di caso d'uso
- ☐ Accettabile (con rischio) nella maggior parte delle organizzazioni - quest'approccio è per tutto e per tutti. L'approccio viene spesso usato su progetti di caso d'uso iniziali nella forma di un processo di requisiti paralleli (ad esempio, il progetto è in esecuzione sia nel vecchio che nel nuovo modo) both the old way and the new way) oppure può essere adottato per nascondere il fatto che gli sviluppatori stanno utilizzando i casi d'uso.
- ☐ Buono per minimizzare l'interruzione dell'organizzazione che per prima adotta o sperimenta casi d'uso. Tutto il mondo continua a vedere il tradizionale SRS, che consente le procedure standard ed i contratti, come l'approccio da usare.

**Contro:**

- ☐ Non ben compreso - è possibile confondersi in presenza sia delle tradizionali istruzioni dei requisiti che del modello di caso d'uso.
- ☐ Questa compresenza comporta la possibilità di due posti in cui restare bloccati nelle attività dei requisiti. È facile restare confusi su quale dovrebbe essere il completo SRS (Software Requirement Specification)
- ☐ Deve essere mantenuto una serie di documenti molto grande.
- ☐ Esistono molti duplicati nel processo di gestione dei requisiti. I tradizionali requisiti software sembrano andare in disuso quando vengono aggiornati i casi d'uso direttamente insieme alle modifiche dei requisiti.
- ☐ È un approccio molto costoso e ad alta manutenzione.

**Esempi**

È utile per le compagnie di sviluppo che usano tecniche di sviluppo guidato del caso d'uso a cui viene fornito un SRS tradizionale come parte del contratto. L'introduzione dei casi d'uso consente alla compagnia di sviluppo di dimostrare la propria comprensione dei requisiti e di distribuire il software in un modo iterativo ed incrementale

Può anche essere utile con l'introduzione di tecniche di caso d'uso in una compagnia che utilizza tecniche di cattura dei requisiti tradizionale ed è restia a passare ad un approccio guidato al caso d'uso. In questo caso l'intenzione è di provare il valore dei casi d'uso all'organizzazione di sviluppo e di ritirare gradualmente l'SRS tradizionale con la crescita della confidenza con i casi d'uso. Può essere il primo passo verso l'approccio "Le funzioni guidano il modello di caso d'uso".

## Il modello di caso d'uso riconcilia più insiemi di requisiti software tradizionali

## Descrizione

"Il modello di caso d'uso è l'interpretazione di SRS formali da fonti multiple e fornisce la specifica di un singolo sistema comune.

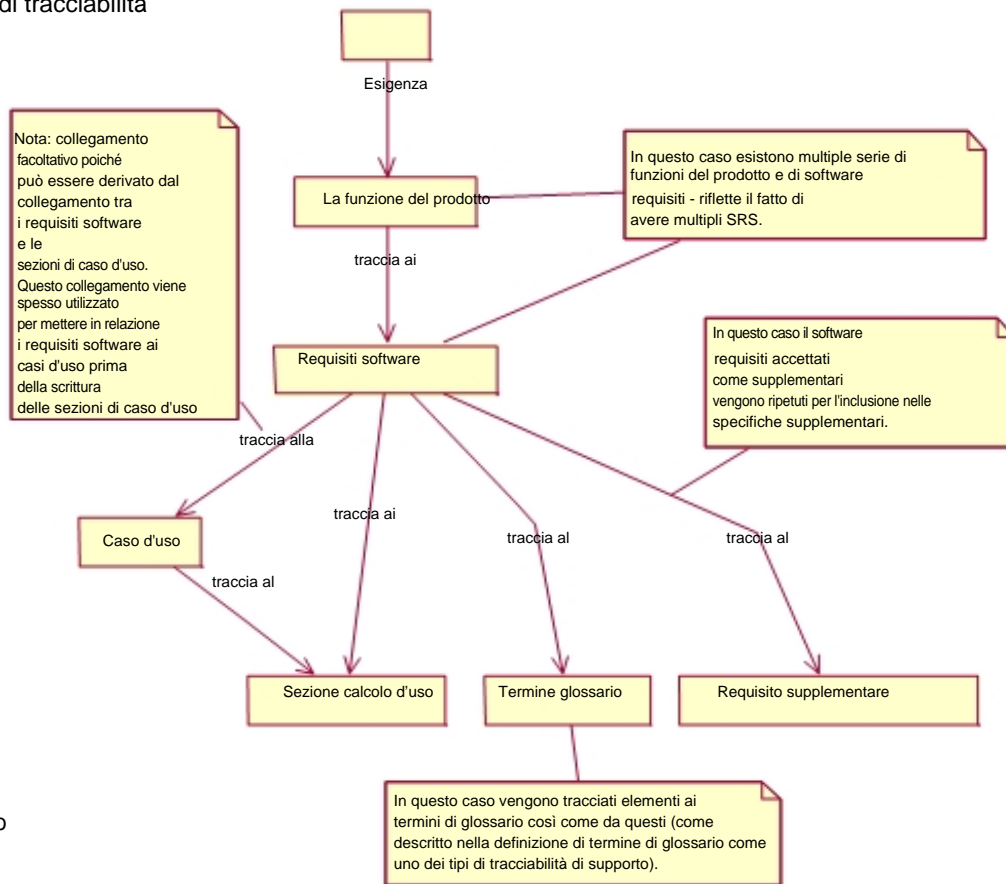
Questa è una variazione dell'assunto che "il modello di caso d'uso è un'interpretazione della specifica dei requisiti software" tranne che per questo caso esistono più SRS tradizionali forniti da altrettante serie di stakeholder. Questa situazione spesso si presenta per case software che sviluppano una singola applicazione per soddisfare i requisiti di molti clienti differenti, indipendenti e nonconnessi. In questo caso il modello di caso d'uso è la vista consolidata degli sviluppatori dei requisiti di sistema e gli SRS individuali sono la vista degli stakeholder dei propri requisiti (senza integrazione o riflessione degli altri requisiti degli stakeholder). La traccia tra le serie di requisiti ed il modello di caso d'uso consente agli sviluppatori di valutare il proprio operato nell'esaminare le esigenze dei diversi stakeholder.

## Caratteristiche

Questa strategia è una variazione dell'approccio precedente del tipo "il modello di caso d'uso è un'interpretazione di SRS". Sono state rilevate solo poche differenze durante la discussione su questo approccio.

Fiducia	Molto Bassa	Come l'approccio del tipo "Il modello di caso d'uso è un'interpretazione di SRS (Software Requirements Specification)".
Responsabilità	Molto Alto	Come l'approccio del tipo "Il modello di caso d'uso è un'interpretazione di SRS (Software Requirements Specification)".
Formalità	Molto Alto	In questo caso vengono preservate tutte le individuali prospettive del cliente nei propri SRS.
Completezza	Molto Alto	Come l'approccio del tipo "Il modello di caso d'uso è un'interpretazione di SRS (Software Requirements Specification)".
Serie di documenti	Molto grande	Come l'approccio del tipo "Il modello di caso d'uso è un'interpretazione di SRS (Software Requirements Specification)".
Focalizzazione	Gestione multipli clienti indipendenti.	In questo caso esistono multiple specifiche del sistema desiderato, riconciliate dal singolo modello di caso d'uso.
Comprensibilità	Media	La focalizzazione di tale approccio è sulla gestione di multiple, indipendenti, possibilmente contraddittorie fonti di requisiti che sono politicamente, geograficamente ed organizzativamente incapaci di lavorare a stretto contatto.
Processo	Tipicamente iterativa e Incrementale.	Come l'approccio del tipo "Il modello di caso d'uso è un'interpretazione di SRS (Software Requirements Specification)".
Stile di sviluppo	Tipicamente Object Oriented	In questo caso gli sviluppatori usano il modello di caso d'uso come il proprio SRS. Vedere "modello di caso d'uso unico"
		In questo caso gli sviluppatori usano il modello di caso d'uso come il proprio SRS per guidare lo sviluppo del software. Vedere "modello di caso d'uso unico"

## Panoramica di tracciabilità



## Tipi di requisito

	Descrizione
Esigenza	Come definito per l'approccio "nessun modello di caso d'uso"
La funzione del prodotto	Come definito per l'approccio "nessun modello di caso d'uso"
Requisiti software	Come definito per l'approccio "nessun modello di caso d'uso"
Caso d'uso	Come definito per l'approccio "modello di caso d'uso unico"
Sezione del caso d'uso	Come definito per l'approccio "modello di caso d'uso unico"
Termini di glossario	Come definito per l'approccio "modello di caso d'uso unico"
	Come definito per l'approccio "modello di caso d'uso unico"
Requisito supplementare	Tutti i requisiti software che vengono applicati all'intero sistema o che non aderiscono facilmente in un caso d'uso.

## Riepilogo di tracciabilità

Collegamento alla tracciabilità	Descrizione
Dall'esigenza alla funzione del prodotto	Come definito per l'approccio "nessun modello di caso d'uso"
Dalla funzione del prodotto al software	Come definito per l'approccio "nessun modello di caso d'uso"
Da SRS al caso d'uso	Come nel caso in cui il modello di caso d'uso è un'interpretazione di SRS
Dal requisito software alla sezione del caso d'uso	Come nel caso in cui il modello di caso d'uso è un'interpretazione di SRS
Dal requisito software ai termini di glossario	Come nel caso in cui il modello di caso d'uso è un'interpretazione di SRS
Dal requisito software ai Specifiche supplementari	I requisiti software devono essere ripetuti nel documento sulle specifiche supplementari che supporta il modello di caso d'uso per consentire la produzione di un singolo, coerente e competente SRS per lo sviluppo che traccia informazioni libere da SRS multipli e individuali dello stakeholder.

## Vantaggi e svantaggi

Questa strategia è una variazione dell'approccio precedente del tipo "il modello di caso d'uso è un'interpretazione di SRS" e comporta gli stessi vantaggi e svantaggi.

Il vantaggio di tale approccio, che lo differenzia dalle altre strategie consiste nella sua capacità di accordarsi con e di preservare i punti di vista di stakeholder indipendenti nel formato dei loro rigorosi e singoli SRS.

Comporta anche lo svantaggio aggiuntivo di generare una serie di documenti anche più grande per la traccia ed il mantenimento.

## Esempi

Una azienda software nel Regno Unito stava sviluppando un sistema di supporto per i broker assicurativi che avrebbe consentito alle compagnie di distribuire in formato elettronico i loro prodotti.

Il progetto era nella regione di 22 stakeholder di cui circa due terzi erano costituiti da compagnie di brokeraggio ed un terzo da compagnie di assicurazione. Tra tutti questi stakeholder esistevano requisiti molto differenti; in molti casi gli assicuratori avevano requisiti completamente contraddittori rispetto ai broker.

Perciò venne decisa la produzione di un SRS per ogni compagnia di stakeholder con il dettaglio degli specifici requisiti e la possibilità di mantenere facilmente le proprie prospettive individuali. Il modello di caso d'uso è stato utilizzato per presentare la visione consolidata del sistema a tutti gli stakeholder. La tracciabilità dagli SRS originali nel modello di caso d'uso ha consentito allo stakeholder di vedere esattamente quali requisiti vengono soddisfatti dal sistema e di convalidare che il sistema è idoneo alle loro esigenze. Ha anche consentito all'azienda software di tracciare il proprio progresso rispetto all'obiettivo di soddisfare l'80% dei requisiti per ogni stakeholder.



Sedi:

Software Rational  
18880 Homestead Road  
Cupertino, CA 95014  
Tel: (408) 863-9900

Rational Software  
20 Maguire Road  
Lexington, MA 02421  
Tel: (781) 676-2400

numero verde: (800) 728-1212

E-mail: [info@rational.com](mailto:info@rational.com)

web: [www.rational.com](http://www.rational.com)

Ubicazione internazionale: [www.rational.com/worldwide](http://www.rational.com/worldwide)

Rational, il logo Rational e Rational Unified Process sono marchi registrati di Rational Software Corporation negli Stati Uniti e/o altri paesi. Microsoft, Microsoft Windows, Microsoft Visual Studio, Microsoft Word, Microsoft Project, Visual C++, e Visual Basic sono marchi o marchi registrati di Microsoft Corporation. Tutti gli altri nomi vengono utilizzati solo per fini di identificazione e sono marchi delle rispettive società. TUTTI I DIRITTI RISERVATI. Realizzato in U.S.A.

Copyright 2002 Rational Software Corporation.

Soggetto a modifiche senza notazione.