

10. *Analisi dei requisiti* (III)

Prof. Paola Barra
a.a. 2022/2023

Documento dei requisiti

Documento dei requisiti

- Il documento dei requisiti è un contratto tra lo sviluppatore e il committente.
- Specifica cosa il prodotto deve fare e quali sono i vincoli che deve soddisfare
- Specifica anche in genere una deadline per la consegna del prodotto

Documento dei requisiti in 5 passi

1. Acquisizione
2. Elaborazione
3. Convalida
4. Negoziazione
5. Gestione

Acquisizione (già visto)

- Interviste
 - Strutturate o non strutturate
- Questionari scritti
 - Scelte multiple,
- Osservazione di futuri utenti al lavoro
- Studio di documenti
- Produzione prototipi
- Casi d'uso
- Con committenti, utenti, responsabili marketing

Elaborazione

- Requisiti espansi e raffinati
- Definizione del documento dei requisiti
 - Basato sull'uso del linguaggio naturale
 - Descrizione del dominio

Struttura del documento dei requisiti

Proposta da Sommerville, e ispirata allo **standard IEEE/ANSI 830-1993** su *requirements documents*

- **Introduzione**
 - Perché il sistema è desiderabile e come si inquadra negli obiettivi piu' generali del cliente
- **Glossario**
 - I termini e i concetti tecnici usati
- **Definizione dei Requisiti funzionali**
 - I servizi richiesti
- **Definizione dei Requisiti non funzionali**
 - I vincoli operativi del sistema, e quelli sul processo di sviluppo
- **Architettura**
 - La strutturazione in sottosistemi (cui riferire i requisiti)

Struttura del documento dei requisiti (2)

- Specifica di System and software requirements
 - Specifica dettagliata dei requisiti funzionali
- Modelli astratti del sistema
 - Modelli formali, o semi-formali: ciascuno illustra un solo punto di vista: controllo, dati, funzioni
- Evoluzione del sistema
 - Previsione di successivi cambiamenti (p. es. di HW, o di requisiti)
- Appendici
 - Individuazione ed eventuale descrizione della piattaforma hardware
 - Requisiti di DataBase
 - Manuale Utente, Piani di Test
- Indici
 - Lemmario: lista di termini, con puntatori ai requisiti che li usano

Glossario

- Definizione dei termini chiave del dominio
- Strumento per comprendere e documentare il dominio
- Ma non solo:
 - Validazione dei requisiti

Esempi di requisiti ben posti

il < sistema > deve < funzionalità > / < proprietà >

- Il sistema deve gestire tutti i registratori di cassa del negozio (non più di 20)
- Il sistema deve stampare la sintesi degli incassi della giornata
- A fine giornata il sistema deve elencare quali sono gli articoli da reintegrare in base alle vendite
- Il tempo di risposta del sistema a qualunque interrogazione deve essere inferiore a 3 secondi

Documento dei requisiti in 5 passi

1. Acquisizione
2. Elaborazione
3. Convalida
4. Negoziazione
5. Gestione

Convalida dei requisiti

Validazione di un documento già strutturato

- Deskcheck
 - Walkthrough
 - lettura “sequenziale” dei documenti
 - Ispezione
 - lettura “strutturata” dei documenti
 - Esempi:
 - tecnica del lemmario
 - Ricerca di rimozioni, distorsioni, generalizzazioni (Noam Chomsky 1975)
- Uso di strumenti di analisi del linguaggio naturale
- Prototipi

Proprietà del documento dei requisiti

■ Difetti da evitare

- Omissioni
 - Mancata presenza di un requisito (incompletezza)
- Inconsistenze
 - Contraddizione fra i vari requisiti o dei requisiti rispetto all'ambiente operativo
- Ambiguità
 - Requisiti con significati multipli
- Sinonimi e omonimi
 - Termini diversi con il medesimo significato e termini uguali con differenti significati
- Presenza di dettagli tecnici
- Ridondanza
 - Può esserci ridondanza, ma solo tra sezioni diverse.

Incompletezze

- *Il sistema dovrà permettere agli utenti di fare delle ricerche per titolo, autore, o ISBN.*
- *Che cosa significa per i CD-ROM?*
 - Possono non avere un ISBN
 - Vale solo per i libri
 - Immaginate se realizzate il sistema con questo requisito senza considerare i CD-ROM.
- Naturalmente non possiamo scrivere requisiti universali, ma possiamo tentare di essere il più possibile completi.

Ambiguità

- Esempio: “Tutti i file sono controllati da un blocco di controllo”
- Interpretazioni:
 1. Un unico blocco di controllo controlla tutti i file
 2. Ogni file ha un suo specifico blocco di controllo
 3. Ogni file è controllato da uno specifico blocco di controllo; alcuni blocchi di controllo possono controllare più di un file
- Il linguaggio naturale è per sua natura ambiguo

Convalida dei requisiti di esempio visti

1. Il sistema software deve fornire un **appropriato** visualizzatore per i documenti memorizzati
 - Appropriato è un termine vago che genera ambiguità
 - Per il resto sembra un requisito ben posto, in forma assertiva: Il sistema deve ...
2. L'utente deve essere in grado di effettuare ricerche sia sull'intero insieme di basi di dati che su **un loro sottoinsieme**
 - Un loro sottoinsieme è ambiguo: uno solo, e in caso quale? Tutti?
 - Per il resto sembra un requisito ben posto, in stile caso d'uso (con gli occhi dell'utente)
3. **Ad ogni nuovo ordine deve essere associato un identificatore unico (Order_ID)**
 - Requisito di basso livello: introduce vincoli implementativi: è necessario ?

Convalida dei requisiti: tecnica del lemmario

- Termini del glossario, con puntatori ai requisiti che li nominano
- Facilita la ricerca di
 - Inconsistenze
 - Sinonimi
 - Omonimi
 - Ridondanze

Convalida dei requisiti: strumenti di analisi lessicale e sintattica

- Analizzano la lista di requisiti in cerca di termini ambigui:
 - Il sistema **può** visualizzare le informazioni di tracciamento dell'ordine.
 - Dopo la consegna, il sistema deve consentire all'utente di inserire le proprie recensioni **o** valutazioni.
 - I prezzi dei prodotti in vendita sono espressi nella valuta **appropriata**

Spazio per migliorare ?

- Il sistema deve gestire tutti i registratori di cassa del negozio (non più di 20)
- Il sistema deve stampare la sintesi degli incassi della giornata
- A fine giornata il sistema deve elencare quali sono gli articoli da reintegrare in base alle vendite
- Il tempo di risposta a qualunque interrogazione deve essere inferiore a 3 secondi

Spazio per migliorare

- Dominio
 - Un negozio contiene al massimo 20 registratori di cassa
 - Sintesi degli incassi = elenco dei totali incassati da ogni registratore di cassa a fine giornata
- Requisiti
 - Il sistema deve gestire tutti i registratori di cassa del negozio
 - Il sistema deve stampare la sintesi degli incassi della giornata
 - Il sistema mantiene un valore di scorta minima per ogni prodotto
 - A fine giornata il sistema deve elencare gli articoli sotto scorta minima

Documento dei requisiti in 5 passi

1. Acquisizione
2. Elaborazione
3. Convalida
4. **Negoziazione**
5. **Gestione**

Negoziazione

- Modifiche alla lista di requisiti
- Divisione in classi (es. MoSCoW)
 - Must have (Requisiti obbligatori)
 - Irrinunciabili per il cliente
 - Should have (Requisiti desiderabili)
 - Non necessari, ma utili
 - Could have (Requisiti opzionali)
 - Relativamente utili, da realizzare se c'è tempo
 - Want to have (Requisiti postponibili)
 - Contrattabili per successive versioni

Requisiti e aspetti contrattuali

- Il documento dei requisiti normalmente precede la stipula del contratto, e ne è parte integrante
- Se alla stipula del contratto non è possibile avere un documento definitivo, è opportuno prevedere di rinegoziare il contratto.
- In caso di gara può essere prodotto dal committente come parte di capitolato tecnico.

Riepilogo

- L'attività di analisi
- Descrizione del dominio
 - Modello del dominio
- Descrizione dei requisiti
 - Documento dei requisiti
- Requisiti funzionali e non funzionali

Attività di scoperta dei requisiti

Attività di scoperta dei requisiti

- Le attività di scoperta dei requisiti fanno corrispondere la definizione di un problema ad una specifica dei requisiti, che può essere presentata come un insieme di
 - Attori
 - Scenari
 - Casi d'uso
- Le attività di scoperta dei requisiti includono
 - Identificazione degli attori
 - Identificazione degli scenari
 - Identificazione dei casi d'uso
 - Identificazione delle relazioni tra attori e casi d'uso
 - Identificazione degli oggetti iniziali dell'analisi
 - Identificazione dei requisiti non funzionali

Identificazione degli attori

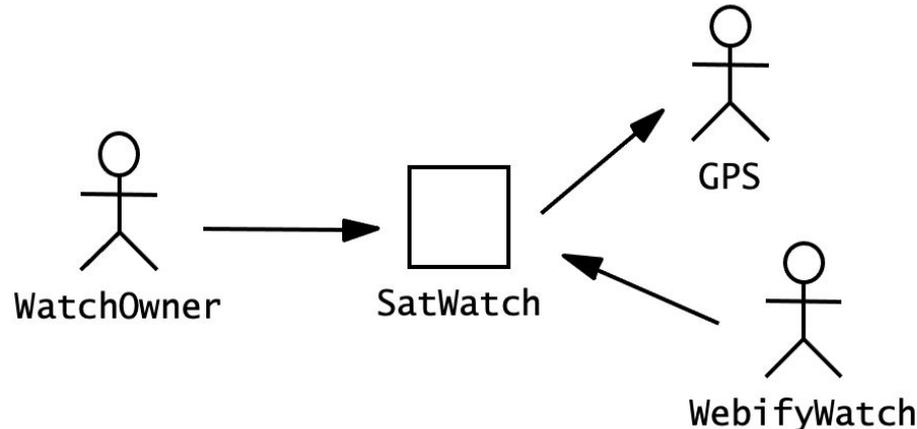
- Gli attori rappresentano entità esterne che interagiscono con il sistema
 - Può essere un utente o un sistema esterno
- Nell'esempio *SatWatch*, il **proprietario** dell'orologio, il satellite **GPS** e il dispositivo seriale **Webify Watch** sono attori
 - Tutti scambiano informazioni con *SatWatch*
 - E' da osservare che tutti hanno interazioni specifiche con *SatWatch*:
 - Il proprietario indossa e guarda l'orologio
 - L'orologio monitora il segnale dal satellite GPS
 - Webify Watch scarica nuovi dati nell'orologio

Attori per il sistema SatWatch

WatchOwner indossa l'orologio, si sposta (eventualmente attraverso diversi fusi) e lo consulta per conoscere l'ora

GPS interagisce con l'orologio per il calcolo della posizione

WebifyWatch aggiorna i dati contenuti nell'orologio per riflettere le modifiche dell'orario

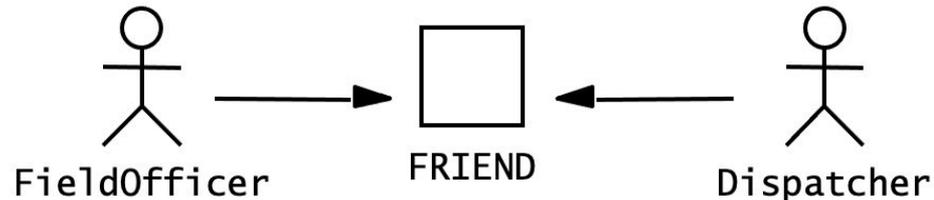


Identificazione degli attori in FRIEND

- Consideriamo l'esempio del sistema **FRIEND**, un sistema informativo distribuito per la gestione degli incidenti
- Include molti attori
 - **FieldOfficer**, gli ufficiali di polizia e dei vigili del fuoco che rispondono ad un incidente
 - **Dispatcher**, l'ufficiale di polizia responsabile di rispondere alle chiamate del 911 e di allocare le risorse ad un incidente
- **FRIEND** supporta ambo gli attori tenendo traccia degli incidenti, delle risorse e dei piani delle attività
 - Inoltre, supporta l'accesso a database multipli, come db dei materiali nocivi e delle procedure delle operazioni di emergenza
 - Gli attori **FieldOfficer** e **Dispatcher** interagiscono attraverso varie interfacce
 - I **FieldOfficer** accedono a FRIEND attraverso un laptop
 - I **Dispatcher** accedono a FRIEND attraverso una workstation

Attori del sistema FRIEND

I *FieldOfficer* non solo hanno accesso a differenti funzionalità, ma usano anche differenti computer per accedere al sistema



Identificazione degli attori

Attori e ruoli sono astrazioni e non necessariamente corrispondono a persone

- La stessa persona può ricoprire il ruolo di **FieldOfficer** e **Dispatcher** in tempi diversi
- Le funzionalità a cui accedono sono sostanzialmente differenti
 - Quindi, i due ruoli sono modellati con due attori differenti

Scoperta dei requisiti e identificazione degli attori

Il primo passo nella scoperta dei requisiti è l'identificazione degli attori

- Serve per definire i confini del sistema e per trovare tutte le prospettive da cui gli sviluppatori necessitano di considerare il sistema

Quando il sistema è inserito in un'organizzazione esistente, solitamente esistono molti attori prima che il sistema sia sviluppato:

- corrispondono a ruoli nell'organizzazione

Scoperta dei requisiti e identificazione degli attori

- Durante la fase iniziale dell'identificazione degli attori, è difficile distinguere gli attori dagli oggetti
 - Ad esempio, un sottosistema **DB** può essere un attore in un frangente e parte del sistema in un altro
- Una volta definito il confine del sistema, non ci sono più problemi nella distinzione
 - Gli attori sono fuori dal confine del sistema, **sono esterni**
 - I sottosistemi e gli oggetti sono all'interno del confine del sistema, **sono interni**
 - Ogni sistema software **esterno** che usa il sistema da sviluppare è un attore

Domande per identificare gli attori

- Quali gruppi di utenti sono supportati dal sistema per eseguire il proprio lavoro?
- Quali gruppi di utenti eseguono le funzioni principali del sistema?
- Quali gruppi di utenti eseguono le funzioni secondarie, come manutenzione e amministrazione?
- Con quali sistemi software e hardware esterni interagirà il sistema?

Attori del sistema FRIEND

Nell'esempio FRIEND, queste domande hanno portato ad una lunga lista di potenziali attori:

- **Vigile del fuoco**, **ufficiale di polizia**, **dispatcher**, **investigatore**, **sindaco**, **governatore**, **db per materiali pericolosi**, **amministratore di sistema** ecc.
- Dopo è necessario consolidare questa lista in un piccolo numero di attori, che differiscono dal punto di vista dell'uso del sistema
 - Ad esempio, un **Vigile del fuoco** ed un **Ufficiale di polizia** possono condividere la stessa interfaccia al sistema, poiché sono entrambi coinvolti con un singolo incidente in loco
 - Un **Dispatcher**, gestisce più incidenti concorrenti e richiede l'accesso ad un maggiore volume di informazioni
 - Il Governatore ed il Sindaco non interagiscono direttamente col sistema ma usufruiscono dei servizi di un operatore addestrato

Identificazione degli scenari

- Uno scenario è una descrizione concreta, mirata e informale di una singola funzionalità del sistema dal punto di vista di un singolo attore
- Gli scenari non possono (e non sono concepiti per) sostituire i casi d'uso, in quanto mirano su specifiche istanze ed eventi concreti
- Gli scenari, comunque, potenziano la scoperta dei requisiti fornendo uno strumento comprensibile agli utenti ed ai clienti

Nome Scenario	<u>warehouseOnFire</u>
Istanze attori partecipanti	<u>bob, alice</u> :FieldOfficer <u>John</u> :Dispatcher
Flusso di eventi	<ol style="list-style-type: none">1. Bob, guidando lungo la strada principale nella sua patrol, osserva del fumo proveniente da un magazzino. Il suo partner, Alice, attiva la funzione “Report Emergency” dal laptotp del FRIEND.2. Alice immette l’indirizzo dell’edificio, una breve descrizione della sua ubicazione (angolo nord-ovest) ed un livello di emergenza. In aggiunta ad un’unità dei vigili del fuoco, richiede diverse unità paramediche poiché la zona non è molto trafficata. Conferma l’input ed attende l’accettazione.3. John, il Dispatcher, è allertato per l’emergenza da un beep dalla sua stazione di lavoro. Rivede le informazioni sottomesse da Alice e accetta il rapporto. Alloca un’unità dei vigili del fuoco e due unità paramediche sul sito dell’Incidente ed invia la stima del tempo di arrivo (ETA) ad Alice.4. Alice riceve l’accettazione e l’ETA.

Stesura degli scenari

- Nella scoperta dei requisiti, sviluppatori e utenti scrivono e rifiniscono una serie di scenari con lo scopo di migliorare la comprensione condivisa di cosa il sistema dovrebbe essere
- Inizialmente, ogni scenario può essere ad alto livello ed incompleto come nel caso del *warehouseOnFire*

Domande per identificare gli scenari

- Quali sono i task che l'attore vuole che il sistema esegua?
- A quali informazioni l'attore accede?
 - Chi crea i dati?
 - Possono essere modificati o rimossi?
 - Da chi?
- Quali modifiche esterne l'attore deve notificare al sistema?
 - Quanto spesso?
 - Quando?
- Quali eventi il sistema deve notificare l'attore?
 - Con quale latenza?

Sorgenti di informazioni per gli scenari

- Gli sviluppatori usano i documenti esistenti sul dominio applicativo per rispondere a tali questioni
- Questi documenti includono **manuali utente** di sistemi precedenti, **manuali delle procedure**, **standard della compagnia**, **note utente**, **interviste con utenti e clienti**
- Gli sviluppatori dovrebbero sempre scrivere gli scenari usando la terminologia del dominio applicativo

FRIEND e scenari

- Nel sistema FRIEND, identifichiamo quattro scenari che abbracciano i tipi di task che ci si aspetta il sistema supporti
 - **warehouseOnFire**: E' individuato un incendio in un magazzino; due ufficiali arrivano sulla scena e richiedono le risorse
 - **fenderBender**: si verifica un incidente automobilistico sull'autostrada senza vittime. Gli ufficiali di polizia documentano l'incidente e gestiscono il traffico mentre i veicoli danneggiati sono portati via dal carro attrezzi
 - **catInATree**: un gatto è bloccato su un albero. Per il recupero del gatto, è richiamato un automezzo dei vigili del fuoco. Poiché l'incidente ha una bassa priorità, il mezzo impiega del tempo per arrivare sul luogo. Nel frattempo, il proprietario del gatto, impaziente, si arrampica sull'albero, cade e si frattura una gamba, richiedendo l'intervento di un'ambulanza
 - **earthQuake**: un terremoto senza precedenti danneggia seriamente edifici e strade, provocando incidenti multipli, innescando l'attivazione di un piano di emergenza nazionale. Il governatore è informato. Le strade danneggiate ostacolano i soccorsi

Identificazione casi d'uso

- Uno scenario è un'istanza di un caso d'uso
 - Un caso d'uso specifica tutti i possibili scenari per una data parte di funzionalità
- Un caso d'uso è iniziato da un attore. Dopo l'avvio, un caso d'uso può interagire anche con altri attori
- Un caso d'uso rappresenta un flusso di eventi completo attraverso il sistema
 - Descrive una serie di interazioni collegate, che sono il risultato dell'avvio

Nome	ReportEmergency
Attori Partecipanti	Iniziato dal FieldOfficer Comunica con il Dispatcher
Flusso eventi:	<ol style="list-style-type: none">1. Il FieldOfficer attiva la funzione "Report Emergency del terminale2. FRIEND risponde presentando una form al FieldOfficer3. Il FieldOfficer riempie la form selezionando il livello di emergenza, tipo, località, breve descrizione della situazione. Il FieldOfficer descrive anche possibili risposte alla situazione di emergenza. Appena finito Il FieldOfficer invia la form4. FRIEND riceve la form e notifica al Dispatcher5. Il Dispatcher rivede le informazioni sottomesse e crea un Incidente nel DB invocando il caso d'uso OpenIncident. Il Dispatcher seleziona una risposta e accetta il rapporto.6. FRIEND visualizza l'accettazione e la risposta selezionata al FieldOfficer
Condizioni di entrata	Il FieldOfficer è loggato in FRIEND
Condizioni di uscita	Il FieldOfficer ha ricevuto un'accettazione e una risposta selezionata dal Dispatcher, OR Il FieldOfficer ha ricevuto una spiegazione indicante perché la transazione non è stata eseguita
Requisiti di qualità	Il rapporto del FieldOfficer è accettato entro 30 secondi La risposta selezionata arriva non più tardi di 30 secondi dopo che è stata inviata dal Dispatcher

Organizzare i casi d'uso

- Inizialmente, gli sviluppatori danno i nomi ai casi d'uso, li allegano agli attori che li iniziano e forniscono una descrizione ad alto livello del caso d'uso
- Il nome di un caso d'uso dovrebbe essere una frase verbale che denota l'attore cosa sta cercando di realizzare
 - **ReportEmergency** indica che un attore sta cercando di fornire un rapporto di emergenza al sistema
 - Non è chiamato **RecordEmergency** poiché il nome dovrebbe riflettere la prospettiva dell'attore, non del sistema
 - Non è nemmeno chiamato **AttemptToReportanEmergency** poiché il nome dovrebbe riflettere l'obiettivo del caso d'uso e non l'attività corrente

Descrizione dei casi d'uso

- Allegare i casi d'uso agli attori che li iniziano consente agli sviluppatori di chiarire i ruoli dei diversi utenti
 - Focalizzandosi su chi inizia ogni caso d'uso, gli sviluppatori identificano nuovi attori che precedentemente sono sfuggiti
- La descrizione di un caso d'uso comporta la specifica di quattro campi:
 - Descrizione delle condizioni di entrata e di uscita
 - Descrizione del flusso di eventi
 - Descrizione dei requisiti di qualità

Esempio: Allocate e Resource

Attori:

- *Field Supervisor*: E' l'ufficiale sul luogo dell'emergenza
- *Resource Allocator*: E' responsabile per l'impegno ed il disimpegno delle risorse gestite dal sistema FRIEND
- *Dispatcher*: Immette, aggiorna, e rimuove Incidenti, Azioni e Richieste nel sistema
 - Provvede anche a chiudere le pratiche sugli incidenti
- *Field Officer*: Produce i rapporti sul luogo dell'emergenza

Esempio: Allocate e Resource

- *Nome caso d'uso*: AllocateResources
- **Attori partecipanti**:
 - Field Officer (Bob e Alice nello Scenario)
 - Dispatcher (John nello Scenario)
 - Resource Allocator
 - Field Supervisor
- **Condizione di ingresso**
 - Il Resource Allocator ha selezionato una risorsa disponibile
 - La risorsa correntemente non è allocata
- **Flusso eventi**
 - Il Resource Allocator seleziona un incidente
 - La risorsa è impegnata sull'incidente
- **Condizione di uscita**
 - Il caso d'uso termina quando la risorsa è impegnata
 - La risorsa selezionata ora è non disponibile per altri incidenti o richieste di risorsa
- **Requisiti speciali**
 - Il Field Supervisor è responsabile della gestione delle risorse

Guida alla stesura dei casi d'uso

- Scrivere i casi d'uso è un'arte
 - Gli analisti imparano con l'esperienza
- Analisti diversi tendono a sviluppare stili diversi
 - Risulta difficile produrre una specifica dei requisiti consistente
- E' possibile usare una guida alla stesura dei casi d'uso

Guida alla stesura dei casi d'uso (2)

- I casi d'uso dovrebbero essere definiti con frasi verbali. Il nome del caso d'uso dovrebbe indicare cosa sta cercando di realizzare l'utente (ReportEmergency, OpenIncident, ecc.)
- Gli attori dovrebbero essere indicati con dei sostantivi (FieldOfficer, Dispatcher, ecc.)
- Il confine del sistema dovrebbe essere chiaro. I passi realizzati dall'attore ed i passi realizzati dal sistema dovrebbero essere chiaramente distinguibili
- I passi dei casi d'uso nel flusso di eventi dovrebbero essere descritti con voce attiva. Ciò rende esplicito chi esegue il passo
- La relazione di causalità tra i passi successivi dovrebbe essere chiara

Guida alla stesura dei casi d'uso (3)

- Un caso d'uso dovrebbe descrivere una transazione utente completa (ad esempio, il caso d'uso **ReportEmergency** descrive tutti i passi tra l'avvio del rapporto di emergenza e la ricezione della notifica avvenuta)
- Le eccezioni dovrebbero essere descritte separatamente
- Un caso d'uso non dovrebbe descrivere l'interfaccia utente del sistema. Ciò distoglie l'attenzione dai passi effettivi realizzati dall'utente
- Un caso d'uso non dovrebbe superare una lunghezza di due o tre pagine. Altrimenti, si devono usare le relazioni **include** e **extend** per decomporlo in casi d'uso più piccoli

Cattivo nome: Che cosa sta cercando di realizzare l'utente?

Nome	Accident
Attore che avvia	Iniziato dal FieldOfficer
Flusso eventi:	<ol style="list-style-type: none">1. Il FieldOfficer notifica un incidente2. E' inviata un'ambulanza.
	<ol style="list-style-type: none">3. Il Dispatcher riceve una notifica quando l'ambulanza arriva sul luogo

Transazione incompleta:
Cosa fa il FieldOfficer dopo che è stata inviata l'ambulanza?

Causalità: Quali azioni comportano la notifica del FieldOfficer?
Voce passiva: Chi invia l'ambulanza?

Rifinire i casi d'uso

- Consideriamo il caso d'uso **ReportEmergency**
 - E' sufficientemente illustrativo per descrivere come FRIEND supporta la creazione dei rapporti di emergenza e per ottenere un feedback generale dagli utenti
 - Tuttavia, esso non fornisce dettagli sufficienti per la specifica dei requisiti
 - ReportEmergency si può ampliare includendo dettagli sul tipo di incidenti noti a FRIEND e interazioni dettagliate che indicano come il Dispatcher notifica il FieldOfficer

Nome	ReportEmergency
Attori Partecipanti	<p>Iniziato dal FieldOfficer Comunica con il Dispatcher</p>
Flusso eventi:	<ol style="list-style-type: none"> 1. Il FieldOfficer attiva la funzione “Report Emergency” del terminale 2. FRIEND risponde presentando una form al FieldOfficer 3. Il FieldOfficer riempie la form selezionando il livello di emergenza, tipo, località, breve descrizione della situazione. Il FieldOfficer descrive anche possibili risposte alla situazione di emergenza. Appena finito Il FieldOfficer invia la form 4. FRIEND riceve la form e notifica al Dispatcher 5. Il Dispatcher rivede le informazioni sottomesse e crea un Incidente nel DB invocando il caso d’uso OpenIncident. Il Dispatcher seleziona una risposta e accetta il rapporto. 6. FRIEND visualizza l’accettazione e la risposta selezionata al FieldOfficer
Condizioni di entrata	Il FieldOfficer è loggato in FRIEND
Condizioni di uscita	<p>Il FieldOfficer ha ricevuto un’accettazione e una risposta selezionata dal Dispatcher, OR Il FieldOfficer ha ricevuto una spiegazione indicante perché la transazione non è stata eseguita</p>
Requisiti di qualità	<p>Il rapporto del FieldOfficer è accettato entro 30 secondi La risposta selezionata arriva non più tardi di 30 secondi dopo che è stata inviata dal Dispatcher</p>

Nome	ReportEmergency
Attori Partecipanti	Iniziato dal FieldOfficer Comunica con il Dispatcher
Flusso eventi:	<ol style="list-style-type: none"><li data-bbox="455 205 1379 238">1. Il FieldOfficer attiva la funzione “Report Emergency” del terminale<li data-bbox="529 244 1856 350">2. FRIEND risponde presentando una form al FieldOfficer. <i>La form include un menu del tipo di emergenza (emergenza generale, incendio, trasporto) e la località, la descrizione dell’incidente, la richiesta di risorse, i campi dei materiali nocivi.</i><li data-bbox="455 357 1779 463">3. Il FieldOfficer riempie la form <i>specificando al minimo il tipo di emergenza e i campi</i> descrizione. Il FieldOfficer descrive anche possibili risposte alla situazione di emergenza <i>e può richiedere risorse specifiche</i>. Appena finito Il FieldOfficer invia la form<li data-bbox="529 469 1483 502">4. FRIEND riceve la form e notifica al Dispatcher <i>con un finestra pop-up</i>.<li data-bbox="455 508 1866 696">5. Il Dispatcher rivede le informazioni sottomesse e crea un Incidente nel DB invocando il caso d’uso OpenIncident. <i>Tutte le informazioni contenute nella form del FieldOfficer sono incluse automaticamente nell’Incident. Il Dispatcher seleziona una risposta allocando le risorse all’Incidente (con il caso d’uso AllocateResources) e notifica il rapporto di emergenza inviando un breve messaggio al FieldOfficer.</i><li data-bbox="529 703 1508 736">6. FRIEND visualizza l’accettazione e la risposta selezionata al FieldOfficer
Condizioni di entrata	...

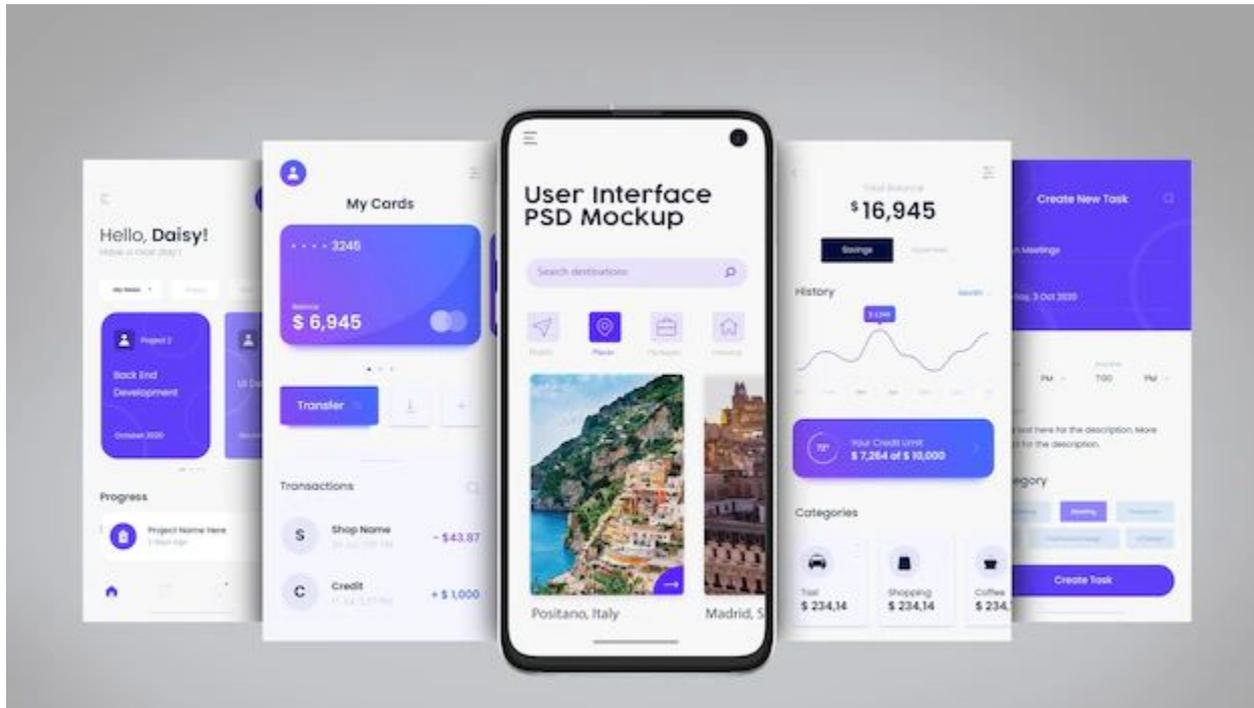
Rifinire i casi d'uso

- L'utilizzo di scenari e casi d'uso per definire le funzionalità del sistema mira a creare requisiti che saranno validati dall'utente prima dello sviluppo
- Appena inizia la progettazione e l'implementazione del sistema, aumenta il costo delle variazioni nella specifica dei requisiti e dell'aggiunta di nuove funzionalità
- Sebbene i requisiti cambiano tardi nel corso dello sviluppo, gli sviluppatori e gli utenti dovrebbero sforzarsi di affrontare i problemi con i requisiti quanto prima

Euristiche per scrivere scenari e casi d'uso

- Usare gli scenari per comunicare con gli utenti e per validare le funzionalità
- Rifinire un singolo scenario per capire le assunzioni sul sistema dell'utente
- Definire scenari non molto dettagliati per definire lo scopo del sistema. Validare con l'utente
- Usare mock-up solo come supporto visuale; il progetto dell'interfaccia utente dovrebbe avvenire come task separato dopo che la funzionalità è sufficientemente stabile
- Presentare all'utente più alternative diverse. Valutare diverse alternative allarga l'orizzonte dell'utente. Generare alternative differenti forza gli sviluppatori a "pensare oltre l'ordinario"
- Dettagliare una parte più ampia quando lo scopo del sistema e le preferenze dell'utente sono ben capiti. Validare con l'utente

mock up



Rifinire i casi d'uso

- L'enfasi di questa attività è la completezza e la correttezza
- Gli sviluppatori identificano le funzionalità non coperte dagli scenari e le documentano rifinendo i casi d'uso o scrivendone nuovi
- Gli sviluppatori descrivono raramente casi e gestione delle eccezioni così come visti dagli attori
- Mentre l'identificazione iniziale dei casi d'uso e degli attori mira a stabilire il confine del sistema, la rifinitura dei casi d'uso porta di volta in volta più dettagli sulle caratteristiche fornite dal sistema e i vincoli ad esse associati

Rifinire i casi d'uso

- Gli aspetti dei casi d'uso che sono inizialmente ignorati e dettagliati durante la rifinitura sono:
 - **Gli elementi manipolati dal sistema** (in *ReportEmergency* sono stati aggiunti dettagli sugli attributi della form del rapporto di emergenza ed il tipo di incidente)
 - **La sequenza a basso livello delle interazioni** tra l'attore ed il sistema (sono state aggiunte informazioni su come il *Dispatcher* genera una notifica selezionando le risorse)
 - **Permessi di accesso** (quale attore può invocare quale caso d'uso)
 - **Eccezioni mancanti** e la loro gestione
 - **Funzionalità comuni** tra i casi d'uso