



APS - Analisi dei requisiti

Created	@June 21, 2024
Tags	APS

3.1 Iterazione 0 e i suoi obiettivi

Nello sviluppo iterativo è comune avere un'iterazione "zero" iniziale, che ha lo scopo di definire la visione del software da realizzare e una stima approssimativa dei costi e dei tempi di sviluppo.

Questo passo iniziale corrisponde alla fase di ideazione di UP.

3.2 Processo: Ideazione

In UP, l'ideazione è appunto il breve passo iniziale che permette di definire la visione del progetto e di ottenere una stima (approssimativa e non affidabile) dell'ordine di grandezza dei costi e dei tempi di sviluppo. A tal fine, l'ideazione richiede *un po'* di esplorazione dei requisiti: per esempio, l'analisi del 10% circa dei requisiti funzionali (casi d'uso), nonché l'analisi dei requisiti non funzionali più critici. Tuttavia, **lo scopo della fase di ideazione non è quello di definire tutti i requisiti**, né di generare una stima o un piano di progetto affidabili.

La maggior parte dell'analisi dei requisiti avviene durante la fase di elaborazione, in parallelo alle prime attività di programmazione di qualità-produzione e di test.

Pertanto, per la maggior parte dei progetti, la fase di ideazione dovrebbe essere relativamente breve, per esempio una settimana o alcune settimane.

<i>L'ideazione in una frase</i>	<i>Il problema principale risolto dall'ideazione, in una frase:</i>
Immaginare la portata del prodotto, la visione e lo studio economico. Le parti interessate hanno un accordo di base sulla visione del progetto, e vale la pena di investire in un'indagine seria?	Le parti interessate hanno un accordo di base sulla visione del progetto, e vale la pena di investire in un'indagine seria?

3.2.1 Elaborati iniziati durante l'ideazione

- Alcuni elaborati vengono iniziati (spesso semplicemente abbozzati) durante l'ideazione
- Non vanno nemmeno creati elaborati se non si ritiene che aggiungano un valore pratico effettivo

3.3 Requisiti Evolutivi



Cos'è un requisito?

*Un **requisito** è una capacità o una condizione a cui il sistema, e più in generale il progetto, deve essere conforme*

Ci sono due tipi principali di requisiti

- I **requisiti funzionali** (comportamentali) descrivono il comportamento del sistema, in termini di *funzionalità* fornite ai suoi utenti. Questi requisiti sono di solito orientati all'uso del sistema e possono essere espressi, per esempio, sotto forma di casi d'uso. I requisiti funzionali comprendono anche gli aspetti relativi alle *informazioni* che il sistema deve gestire.
- I **requisiti non funzionali** (tutti gli altri requisiti) non riguardano le specifiche funzioni del sistema, ma sono relativi a proprietà del sistema nel suo complesso, come per esempio sicurezza, prestazioni, scalabilità, usabilità ecc.

I problemi sorgono quando i requisiti funzionali non sono indicati con precisione. Requisiti ambigui possono essere interpretati in modi diversi da sviluppatori e utenti. In linea di principio, i requisiti dovrebbero essere **completi** e **coerenti**.

Tuttavia, i requisiti non funzionali possono essere più critici dei requisiti funzionali. In caso contrario, il sistema potrebbe risultare inutilizzabile. Questo perché possono influire sull'architettura complessiva di un sistema piuttosto che sui singoli componenti. Inoltre, possono essere molto difficili da stabilire con precisione e requisiti imprecisi possono essere difficili da verificare.

Metriche per specificare i requisiti non funzionali: Velocità, Dimensione, Facilità d'uso, Affidabilità, Robustezza, Portabilità

Quali sono i modi validi per trovare requisiti: Interviste con i clienti, Scrivere casi d'uso con i clienti, Workshop dei requisiti a cui partecipano sia sviluppatori che clienti, Gruppi di lavoro con rappresentanti dei clienti.

Linee guida per la scrittura dei requisiti

- Ideare un formato standard ed utilizzarlo per tutti i requisiti
- Evitare il gergo informatico
- Includere una spiegazione (razionale) del motivo per cui è necessaria una disposizione.

La scrittura dei requisiti può avvenire per notazioni grafiche, linguaggio naturale strutturato e specifiche formali.

3.4 Casi d'uso

I casi d'uso sono storie scritte, ampiamente utilizzati per scoprire e registrare i requisiti. Essi influenzano molti aspetti di un progetto, compresa l'analisi e la progettazione orientata agli oggetti.

N.B.: I casi d'uso NON sono elaborati orientati agli oggetti bensì testo

- Un attore è qualcosa o qualcuno dotato di comportamento

- Uno scenario (o istanza del caso d'uso) è una sequenza specifica di azioni e iterazioni tra il sistema e alcuni attori
- Un caso d'uso è una collezione di scenari correlati, sia di successo che di fallimento, che descrivono un attore che usa un sistema per raggiungere un obiettivo specifico

Modello dei Casi d'Uso

Il Modello dei Casi d'Uso è l'insieme di tutti i casi d'uso scritti.

- La modellazione dei casi d'uso è innanzitutto un atto di scrittura di testi, non di disegno di diagrammi
- Il modello dei casi d'uso può includere, opzionalmente, un diagramma UML dei casi d'uso

I casi di uso sono requisiti, soprattutto requisiti funzionali o comportamentali, che indicano che cosa deve fare il sistema. Possono essere utilizzati anche per altri tipi di requisiti, soprattutto se questi sono fortemente correlati a un caso d'uso.

Un caso definisce un contratto relativo al comportamento di un sistema.

I casi d'uso possono essere scritti usando diversi formati e livelli di formalità:

1. Formato breve
2. Formato informale
3. Formato dettagliato

I casi d'uso dovrebbero essere scritti in modo conciso, lo stesso tempo in modo completo. Inoltre, si deve specificare che cosa deve fare il sistema (comportamento o requisiti funzionali) senza decidere come lo farà (progettazione), ovvero **scrivere casi d'uso a scatola nera** (Figura 3.1).

Infine ci si deve concentrare su utenti o attori del sistema, chiedendo quali sono i loro obiettivi e le situazioni tipiche, per comprendere ciò che l'attore considera un risultato di valore.

Stile a scatola nera	Non a scatola nera
Il Sistema registra la vendita.	Il Sistema memorizza la vendita in una base di dati. ... o (ancora peggio):
Il Sistema esegue un'istruzione SQL INSERT per la vendita...	

Figura 3.1: Scrivere casi d'uso a scatola nera e non a scatola nera.

Livello dei casi d'uso

I casi d'uso possono essere scritti a livelli diversi

1. **Livello di obiettivo utente:** è il più interessante nell'analisi dei requisiti
2. **Livello di sotto-funzione:** utili per mettere a fattor comune parti di casi d'uso e/o per descrivere interazioni di dettaglio
3. **Livello di sommario:** un caso d'uso a questo livello comprende più casi d'uso a livello di obiettivo utente (utile per la loro identificazione, e per comprendere il loro contesto)

Punti di forza e usi

- Enfatizzano gli obiettivi degli utenti
- Decompongono la funzionalità del sistema in un insieme di compiti discreti (Divide et impera)
- Facile da capire per gli utenti
- Può essere riutilizzato per la documentazione utente
- I casi di test possono essere derivati direttamente dai casi d'uso
- Indipendente dalla tecnologia di implementazione

3.4.1 Tipi di attori

Un attore è qualcosa o qualcuno dotato di comportamento - Anche il sistema in discussione (SuD) stesso è considerato un attore, quando ricorre ai servizi di altri sistemi.

1. **Attore primario:** raggiunge degli obiettivi usando il SuD
2. **Attore finale:** vuole che il SuD sia utilizzato affinché vengano raggiunti dei suoi obiettivi
3. **Attore di supporto:** offre un servizio (per esempio informazioni) al SuD
4. **Attore fuori scena:** ha un interesse nel comportamento del caso d'uso SuD, ma non è un attore primario, finale o di supporto

3.4.2 Come trovare i casi d'uso

1. **Scegliere il confine di sistema**
E' importante capire cosa sta fuori dai confini del sistema. Una volta identificati gli attori esterni, i confini diventano più chiari.
2. **Identificare gli attori primari**
3. **Identificare gli obiettivi per ogni attore primario**
4. **Definire i casi d'uso che soddisfano questi obiettivi**

3.4.3. Verificare l'utilità dei casi d'uso

- Test del capo
- Test EBP (Elementary Business Process)
- Test della Dimensione

3.4.3.1 Test del capo

Il vostro capo sarebbe felice se voi faceste sempre la stessa identica azione tutto il giorno?

- *Se non lo è il caso d'uso non è fortemente mirato ottenere risultati il cui valore sia misurabile*

Ciò non significa che bisogna sempre ignorare i casi d'uso che non superano il test del capo.

3.4.3.2 Il test EBP

Un processo di business elementare è un'attività svolta da una persona in un determinato tempo e luogo, in risposta a un evento di business, che aggiunge un valore di business misurabile e lascia i dati in uno

stato consistente

Concentrarsi sui casi d'uso che corrispondono a degli EBP. La definizione non va presa in modo troppo letterale.

3.4.3.3 Test della Dimensione

Un caso d'uso è raramente costituito da una singola azione o passo; normalmente comprende diversi passi, e nel suo formato dettagliato spesso richiede da 3 a 10 pagine di testo.

Un errore comune nella modellazione dei casi d'uso è definire un caso d'uso a sé formato da un singolo passo, all'interno di una sequenza di passi correlati.

Violazioni ragionevoli dei test:

Attività secondarie e piccoli passi possono diventare casi d'uso per evitare le ripetizioni del testo, in questo caso i tre test possono essere violati