**DOMINIO DEL PROBLEMA**

è come la prima intervista al cliente che spiega “il mondo” del problema, l’ambito in cui dovrà essere sviluppato il software, cosa farà il software in generale

**SCENARI**

esempi concreti di quello che l’utente può fare, con più dettagli possibili (sulle informazioni che l’utente inserisce così da capirne anche il formato, sulla schermata che l’utente vedrà, sui dati coinvolti …) in modo da poter interagire con utente e capire i requisiti

può essere descrizione molto lunga che spiega utente dove si trova mano a mano e cosa fa oppure tanti piccoli scenari ognuno relativo a una particolare situazione (in ordine di sequenza temporale)

(es: all’inizio utente che vuole acquistare accede e si trova sull’homepage, sceglie libro Harry Potter, compra il libro usando Mastercard …)

[nome scenario, attori partecipanti, flusso di eventi]

**REQUISITI FUNZIONALI**

“il sistema permette all’utente di …”

quello che si può fare con il software (utente/tecnico/amministratore) magari divisi in varie categorie oppure in base agli attori

(es: funzionalità per la registrazione/gestione account)

(es: il sistema deve permette a un utente registrato l’acquisto di un libro)

**REQUISITI NON FUNZIONALI**

devono essere verificabili, cioè si possono creare dei test per verificarli

(es: facilità d’uso …)

si deve specificare cosa deve essere implementato e come

-usabilità (sistema responsive, grafica semplice …)

-affidabilità(sistema capace di rispondere a tanti input, anche in caso di errori)

-performance(ad esempio tot accessi simultanei…)

-manutenibilità(modo di gestire il sistema, ad esempio le varie classi, l’utilizzo di design pattern…)

-implementazione(linguaggi usati, Java, HTML, CSS…)

-sicurezza(gestione di dati sensibili criptati, protocollo HTTPS per garantire comunicazioni sicure client-server…)

-legal(utente deve leggere e comprendere le policy della piattaforma, altrimenti potrebbe essere bannato..)

**TARGET ENVIROMENT**

scrivere che tipo di software è (web), quali linguaggi saranno usati, come gestiamo i database …

**DEADLINES**

(si possono seguire scadenze di e-learning)

-Problem statement

-Requisiti e casi d’uso

-Requirements analysis document

-System design document

-Specifica delle interfacce dei moduli del sottosistema da implementare

-Piano di test di sistema e specifica dei casi di test per il sottosistema da implementare