

Scuola Politecnica e delle Scienze di Base

Corso di *Laurea Magistrale* in
Ingegneria Informatica



**Ideazione, Requisiti Evolutivi,
Casi d'Uso, Storie Utente**

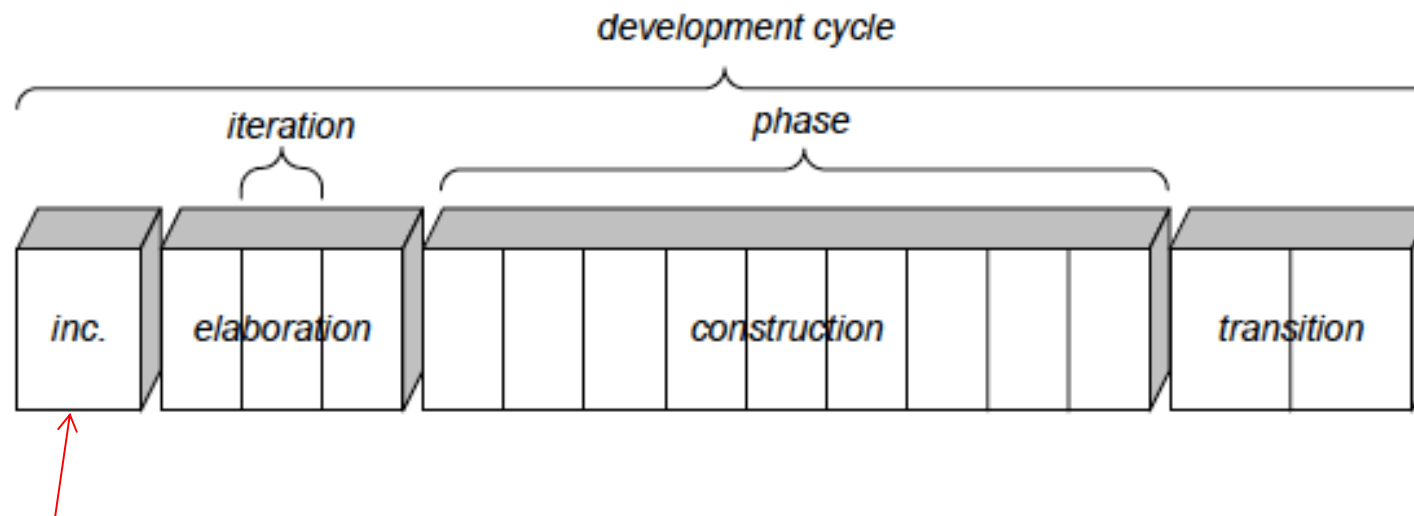
Cap. 5,6,7,8,9 Larman

Corso PSSS- Prof. Fasolino - 2019

Sommario

- Fase di Ideazione
- Confronto Requisiti evolutivi e a cascata
- Tipi di requisiti- FURPS+
- Elaborati dei Requisiti in UP
- Elaborati dei Requisiti in Scrum
- Casi d'Uso
- Altri Requisiti
- Storie Utente

Cosa è l'Ideazione



Ideazione



Cosa è l'ideazione

- Ogni progetto parte con una fase in cui ci si chiede:
 - Quale è la Visione e lo studio economico del progetto ?
 - Il progetto è fattibile?
 - Comprare e/o costruire?
 - Stima dei costi approssimativa (ordine di grandezza)
 - Procediamo o ci fermiamo?
- Pertanto l'Ideazione richiede un po' di esplorazione dei requisiti, ma non deve durare troppo (1 o 2 settimane)
 - La maggior parte dell'analisi dei requisiti sarà svolta in fase di Elaborazione, in parallelo con il primo sviluppo e test del codice.

Scopo dell'Ideazione

- Lo scopo è immaginare la portata del prodotto, la Visione, e lo studio economico.
- Alla fine di questa fase, si sarà deciso se vale la pena continuare ad esplorare oltre (in fase di Elaborazione).
 - *Analogia con la fase preliminare all'avvio di trivellazioni esplorative in campo petrolifero.*
- Può anche comprendere il primo Workshop dei requisiti e la pianificazione della prima iterazione

Elaborati iniziati in Ideazione

Artifact ¹	Comment
Vision and Business Case	Describes the high-level goals and constraints, the business case, and provides an executive summary.
Use-Case Model	Describes the functional requirements, and related non-functional requirements.
Supplementary Specification	Describes other requirements.
Glossary	Key domain terminology.
Risk List & Risk Management Plan	Describes the business, technical, resource, schedule risks, and ideas for their mitigation or response.
Prototypes and proof-of-concepts	To clarify the vision, and validate technical ideas.
Iteration Plan	Describes what to do in the first elaboration iteration.
Phase Plan & Software Development Plan	Low-precision guess for elaboration phase duration and effort. Tools, people, education, and other resources.
Development Case	A description of the customized UP steps and artifacts for this project. In the UP, one always customizes it for the project.

Elaborati in Ideazione

- Molti elaborati prodotti in questa fase saranno solo abbozzati e saranno completati nelle fasi successive
- Ad es. Il Modello dei Casi d'Uso conterrà solo i nomi di tutti gli Attori e dei Casi d'Uso
- Un 10% dei casi d'uso potrà ad es. essere dettagliato


Ideazione e SCRUM

- In Scrum non ci sono le fasi, ma è previsto comunque un passo iniziale detto **Envisioning**
- È simile all'Ideazione, definisce la visione del prodotto da iniziare e un piano di lavoro approssimativo
- Viene creata la versione iniziale del **Product Backlog**

Requisiti Evolutivi

(Cap. 5 Larman)

Requisito

- **Definizione:** È una capacità richiesta al sistema, oppure una condizione a cui esso deve essere conforme.
 - Capacità di gestire informazioni, eseguire operazioni, in tempi massimi prestabiliti, etc.
- **Requisiti Funzionali:** descrivono il comportamento del sistema in termini di funzionalità offerte.
 - Espressi mediante Casi d'Uso
- **Requisiti Non Funzionali:** non riguardano  funzionalità, ma sono relativi a proprietà del sistema
 - Sicurezza, prestazioni, scalabilità, usabilità,...

Confronto fra Requisiti Evolutivi e a Cascata

- Nell'approccio a cascata si richiede di analizzare e specificare *tutti i requisiti prima* che inizi lo sviluppo.
 - Basato su assunzione che un progetto software abbia bassi tassi di cambiamento!
 - Statistiche invece indicano un tasso di cambiamento del 25% medio.
- Nell'approccio Evolutivo (in UP, Scrum, XP, ...) si inizia a scrivere codice e a testare *molto prima di aver completato* l'analisi e la specifica di tutti i requisiti!

Come trovare i Requisiti

- La best practice di UP di gestire i requisiti richiede:
 - *Un approccio sistematico per trovare, documentare, organizzare e tracciare i requisiti che cambiano di un sistema*
- *Alcune tecniche:*
 - *Scrivere i Casi d'uso con i clienti*
 - *Workshop dei requisiti (con sviluppatori e clienti)*
 - *Gruppi di lavoro sviluppatori-clienti*
 - *Dimostrazioni ai clienti ad ogni iterazione-> feedback*

Gestione dei Requisiti

- Classificare i requisiti usando modelli noti. Ad es. ISO 9126, SEI, etc..
- Modello **FURPS** + (*Functional, Usability, Reliability, Performance, Supportability*)
- + sta per altri requisiti, complementari e secondari, come:
 - Vincoli di progetto (limiti risorse, linguaggi, tool, hardware)
 - Interoperabilità (interfacciamento con altri sistemi)
 - Operazionali (gestione del sistema in contesto operativo)
 - Fisici (es. vincoli su dimensioni fisiche per l'hardware)
 - Legali ...(licenze)

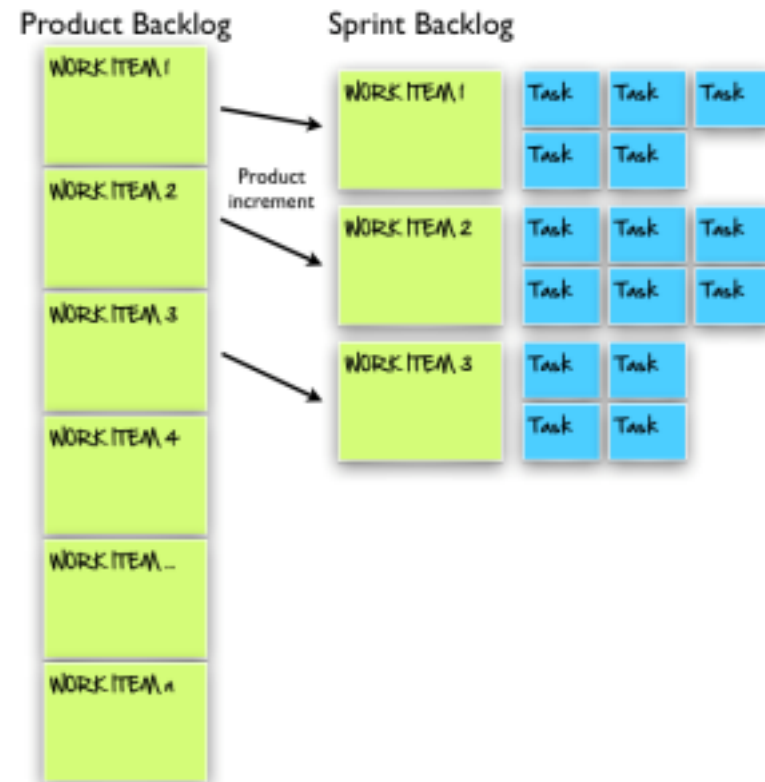
Requisiti ed Elaborati di UP



- Modello dei Casi d'Uso
 - Specifiche Supplementari (req. Non funz., licenze..)
 - Glossario (con ruolo di Dizionario dei Dati)
 - Visione (Requisiti ad alto livello e studio economico)
 - Regole di Business (regole di dominio che il software dovrà rispettare)
-
- Gli elaborati possono essere descritti in una Wiki, o mediante template (v. RUP IBM).


Requisiti ed elaborati di Scrum

- In Scrum i requisiti sono riportati in un **Product Backlog** che riporta elenco funzionalità e requisiti non ancora sviluppati.
- Ad ogni Sprint, le voci del Backlog vengono aggiunte/ rimosse/ raffinate e si crea lo Sprint Backlog.



Requisiti in Scrum

- In genere espressi mediante Storie Utente

ToDo List 		
Story	Estimation	Priority
As a user I want to be able to reset my password	1	1
As a user I want to edit items	3	2
As a user I want to export data	2	3
As an administrator I want to define KPI's for my sales team	4	4
As a user I want to view my data on mobile	5	5
As an administrator I want to send alerts when new leads come in	2	6
As a user I want to create a report of my data	5	7
As a user I want to update my reminder settings when a date is added	3	8
As a user I want filtering enhancements	4	9
As an administrator I want to configure views of data	5	10
Total	34	



Un esempio di riferimento: il Sistema POS NextGen

Un sistema usato per registrare le vendite e gestire i pagamenti al Supermercato



Sistema POS NextGen


- È un sistema fatto di software ed hardware (computer e lettore di codice a barre).
- Si interfaccia con altri sistemi software esterni (Sistema per l'inventario o la contabilità)
- Deve essere tollerante ai guasti e permettere sempre di portare a termine il pagamento (al limite in contanti)
- Deve supportare diversi tipi di interfacce lato client (un browser web thin-client, un PC con GUI grafica tipo Java Swing, dispositivi touch screen, palmari, etc..)
- Deve essere commercializzato a vari clienti con diverse esigenze e quindi deve essere personalizzabile

Casi d'Uso (Cap. 7 Larman)



*Il primo passo indispensabile per ottenere dalla vita
ciò che si vuole: decidere cosa si vuole*

Ben Stein

Casi d'Uso

- **Definizione:** I casi d'uso sono storie scritte, testuali, di qualche attore che usa il sistema per raggiungere degli obiettivi. 
 - Essi influenzano l'analisi e progettazione OO di un sistema
- **Esempio:** *Elabora Vendita*
- Un cliente arriva alla cassa con gli articoli. Il cassiere usa il POS per registrare ogni articolo. Il sistema mostra il totale e i dettagli di ogni articolo. Il cassiere inserisce informazioni sul pagamento, che il sistema convalida e registra. Il sistema aggiorna l'inventario. Il cliente riceve una ricevuta e se ne va con gli articoli acquistati.

Alcune definizioni preliminari


- Un **Attore** è qualcosa o qualcuno dotato di comportamento, come una persona (con un ruolo), un sistema informatico, o una organizzazione. 
- Uno **Scenario** (o istanza di caso d'uso) è una sequenza specifica di azioni ed interazioni fra il sistema ed alcuni attori. 
 - Descrive un percorso attraverso il caso d'uso
 - Es. Scenario di acquisto che va a buon fine, pagando in contanti
 - Es. Scenario di fallito acquisto, siccome non è stata concessa l'autorizzazione ad un pagamento con carta di credito.

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.
- Es. Caso d'Uso Gestisci Restituzione Prodotto
- Scenario di successo:
 - Il cliente arriva alla cassa con alcuni articoli da restituire. Il cassiere usa il POS per registrare gli articoli... il cliente riceve il rimborso.
- Scenari alternativi:
 - Se il cliente aveva pagato con carta e il rimborso su carta viene respinto, il cassiere rimborsa in contanti.
 - Se il codice del prodotto non viene rilevato, allora il cassiere inserisce a mano ...
 - Etc.

Modello dei Casi d'Uso

- È l'insieme dei casi d'uso scritti.
- Può includere opzionalmente un Diagramma dei Casi d'uso che mostra attori, casi d'uso e relazioni fra essi. 
- È utile anche come Diagramma di Contesto del sistema e del suo ambiente.
- Il vantaggio dei casi d'uso è nella loro semplicità (rispetto ad altri metodi di analisi) e nel porre in primo piano i bisogni dell'utente. Inoltre sono usabili a vari livelli di dettaglio

Vari tipi di attori


- **Attore Primario:** 
 - Usa direttamente i servizi del sistema (SuD) per raggiungere gli obiettivi utente (es. cassiere al POS)
- **Attore Finale:** vuole che il sistema venga usato per raggiungere i suoi obiettivi (es. cliente al POS) 
- **Attore di Supporto:**
 - Offre un servizio al SuD; Es. servizio informatico di autorizzazione pagamenti
 - È importante per chiarire interfacce con l'esterno e protocolli
- **Attore Fuori Scena** 
 - Ha interesse nel comportamento del caso d'uso,
 - È necessario identificarlo, per essere certi che tutti gli interessi saranno soddisfatti

Tre Formati per i Casi d'Uso

- **Formato Breve**


- Riepilogo conciso, di un solo paragrafo (v. es. *Elabora Vendita*)
- Va usato nella fase iniziale dei requisiti, per capire rapidamente l'argomento.

- **Formato Informale**




- Più paragrafi, informali, relativi a vari scenari
- Es. Gestisci restituzione 
- *Scenario di successo*: Un cliente arriva alla cassa con articoli da restituire, il cassiere usa il POS
- *Scenari alternativi*:
 - se il cliente aveva pagato con carta di credito,...
 - Se il codice prodotto non viene trovato, il cassiere inserisce..

Tre Formati per i Casi d'Uso

- **Formato Dettagliato**

- Tutti i passi e le varianti sono scritte nel dettaglio, con pre-condizioni, etc....
- Deve essere usato dopo che tutti i casi d'uso saranno stati identificati, e un 10% di essi con maggiore valore verrà scritto in formato dettagliato 
- Vedere Template Standard per i Casi d'uso

Template per Casi d'uso dettagliati

- Nome del Caso d'uso (inizia con un verbo) 
- Portata (il sistema che si sta progettando)
- Livello (Obiettivo utente o sottofunzione) 
- Attore primario
- Parti interessate e interessi
- Pre-condizioni 
- Garanzia di successo (cosa è garantito in caso di successo del Caso d'uso)
- Scenario Principale
- Estensioni (Scenari alternativi, sia di successo che di fallimento)

Esempio del POS- Elabora Vendita

- **Portata:** Applicazione POS NextGen
- **Livello:** Obiettivo Utente
- **Attore Primario:** Cassiere
- **Parti interessate:**
 - Cassiere (vuole inserimento dati rapido e preciso)
 - Addetto alle vendite (vuole che le commissioni sulle vendite siano aggiornate)
 - Cliente (vuole fare acquisti in modo rapido, vuole dettaglio spesa, e prova d'acquisto)
 - Azienda (vuole registrare le transazioni, registrare i pagamenti da riscuotere dal Servizio Pagamenti, ...)
 - Direttore (vuole poter risolvere rapidamente i problemi del cassiere..)
 - Enti Fiscali (Vogliono riscuotere le tasse sui pagamenti)
 - Servizio autorizzazione Pagamenti (vuole ricevere le richieste di autorizzazione pagamenti...)

- **Pre-condizioni:** Il Cassiere è autenticato ed identificato
- **Garanzia di Successo:** La vendita viene salvata. Le imposte sono calcolate. Contabilità e inventario sono aggiornati. Le commissioni sono calcolate. Le approvazioni delle autorizzazioni di pagamento sono registrate.
- **Scenario Principale di successo (Flusso Base)**
 1. Il cliente arriva alla cassa POS con gli articoli da acquistare
 2. Il Cassiere inizia una nuova vendita
 3. Il Cassiere inserisce il codice di un articolo
 4. Il sistema registra la riga di vendita dell'articolo e mostra descrizione, prezzo dell'articolo e totale parziale
 5. Il Cassiere ripete i passi 3-4 fino a che indica che ha terminato
 6. Il sistema mostra il totale con le imposte
 7. Il Cassiere mostra il totale al cliente e chiede il pagamento
 8. Il Cliente paga e il sistema gestisce il pagamento
 9. Il sistema registra la vendita completata, invia informazioni sulla vendita al Sistema esterno di Contabilità ed Inventario
 10. Il Sistema genera la ricevuta
 11. Il Cliente va via

Flussi alternativi

3.a: Codice identificativo non valido

1. Il sistema segnala l'errore

2a. il cassiere inserisce il codice manualmente

2b. Il cassiere chiede al Direttore di intervenire
per inserire manualmente il prezzo

2c. Il Cassiere esegue Ricerca Prodotto per
ottenere il vero prezzo

3.b: Ci sono più articoli della stessa categoria (es. 5 hamburger)

1. Il Cassiere inserisce codice e quantità

3.6 a: Il Cliente chiede di eliminare un articolo

1. Il Cassiere inserisce il codice da rimuovere

2. Il sistema elimina l'articolo e aggiorna il conto parziale


.....

Sul libro sono presenti ben 5 pagine di specifiche dettagliate di questo caso d'uso

Descrivere le Estensioni o flussi alternativi del caso d'uso

- Lo scenario base di uno UC è quello *felice*, di successo. Non contiene alcuna condizione o diramazione.
- Il comportamento condizionale e delle alternative è in genere descritto nella sezione delle Estensioni
 - Per rendere la descrizione più semplice
- Invece di riportare tutto lo scenario alternativo, si può riportare solo l'estensione a partire da un certo punto dello scenario di successo
- L'estensione sarà etichettata con una lettera, dopo il numero del punto dello scenario che viene sostituito
- Es. Dal punto 3 -> punti 3a, 3b, 3c, ...
- L'estensione riporta prima la condizione e poi la reazione

Altre regole

- Al termine della gestione di una estensione, in genere lo scenario si fonde di nuovo con lo scenario principale del caso d'uso. 
- È anche possibile che in un certo punto dello scenario venga eseguito un altro caso d'uso completo.
- Lo si può indicare indicando il nome del Caso d'uso eseguito con una sottolineatura
- Es. Il cassiere **esegue** Aiuto Ricerca Prodotto per ottenere il codice di un prodotto


Linee Guida per descrivere i casi d'uso

- **Scrivere in uno stile essenziale**, senza riferimento all'Interfaccia Utente, concentrandosi sullo scopo.
 - Es. Stile essenziale: l'Amministratore si identifica
 - Stile non essenziale: l'Amministratore inserisce Id e Password nella finestra di dialogo
 - Casi d'uso non essenziali potranno servire in fase di progettazione, non di analisi
-
- **Scrivere casi d'uso a scatola nera (ossia il cosa, non il come)**
 - Es. Il sistema registra la vendita
 - Es. Non a scatola nera: il sistema registra la vendita nella Base di Dati (o peggio, fa una query SQL)


Linee Guida per descrivere i casi d'uso

- Adottare il punto di vista dell'attore e **dell'attore-obiettivo** (Ivar Jacobson)
 - Ogni caso d'uso deve corrispondere ad un obiettivo di un attore
 - Un caso d'uso è un insieme di istanze di caso d'uso, ove ogni istanza è una sequenza di azioni che il sistema esegue per produrre un risultato osservabile e di valore per uno specifico attore
 - È un modo per concentrarsi su ciò che gli utenti vogliono realmente dal software, invece di elencare solo caratteristiche e funzioni del software (v. SRS)

Linea Guida per trovare i casi d'uso

- 1) Scegliere i Confini del Sistema 
 - È un'applicazione software, o un sistema fatto di software + hardware, o tutto questo+ le persone che lo usano?
- 2) Identificare gli Attori primari
- 3) Identificare gli Obiettivi degli attori primari
- 4) definire i casi d'uso che soddisfano gli obiettivi degli utenti.

1. Scegliere i Confini del sistema

- Stabilire ciò che è interno al sistema e ciò che è esterno.
- Può essere utile partire dagli attori, che sono esterni al sistema, per definizione. 
- Ad esempio, nel caso del sistema POS, sono esterni il Cassiere e il Sistema di autorizzazione dei pagamenti.

2 e 3: Trovare gli attori primari ed i loro obiettivi

- Oltre agli attori primari “ovvi” chiedersi anche:
- Chi avvia e arresta il sistema?
- Chi si occupa di gestire gli utenti e la sicurezza?
- Esiste un processo di monitoraggio che riavvia il sistema in caso di errore?
- Il tempo è un attore? Nel senso che il sistema esegue operazioni in risposta ad eventi temporali?
- Chi viene avvisato in caso di guasti?
- ...

Costruire una Tabella Attori-Obiettivi

Attore	Obiettivo	Attore	Obiettivo
Cassiere	Elaborare le vendite Gestire le restituzioni ...	Amministratore di sistema	Aggiungere Utenti Modificare utenti Eliminare utenti Gestire sicurezza...
Direttore	Avviare il sistema Arrestare il sistema...	Sistema di Analisi delle Vendite	Analizzare dati sulla vendita ...
...			

Ad ogni obiettivo può corrispondere un caso d'uso


Definire il Caso d'uso

- Assegnare al caso d'uso lo stesso nome dell'obiettivo utente.
- Linea Guida: il nome del caso d'uso deve iniziare con un verbo.
- Una eccezione per gli obiettivi CRUD (create, retrieve, update, delete) che possono diventare un solo caso d'uso dal nome Gestisci , es. gestisci utente.


Linee Guida: Verificare l'Utilità dei Casi d'Uso

- **1) Il Test del Capo**
 - Il capo sarebbe felice di un caso d'uso Login?
- **2) il Test EBP (Elementary Business Process)**
 - Un processo di business elementare è una attività svolta da una persona, in una sola sessione di lavoro, in risposta ad un evento che aggiunge un valore di business misurabile e lascia i dati in uno stato consistente (es. approva un credito, o Stabilisci il prezzo per un ordine)
 - Concentrarsi su UC che corrispondono a EBP
- **3) Il Test della dimensione:**
 - Un caso d'uso raramente è costituito da una singola azione o singolo passo
 - Nella forma dettagliata un caso d'uso può occupare 3/10 pagine di testo

Tre Livelli dei Casi d'Uso

- **Livello di Obiettivo Utente**
 - È il caso d'uso che permette all'utente di raggiungere un proprio obiettivo (es. *Gestire restituzione prodotto*)
- **Livello di sotto-funzione** 
 - Può servire a descrivere un singolo passo nell'utilizzo del sistema (Es. *Effettuare il Login*)
 - Non serve da solo a raggiungere un obiettivo utente
 - Servono solo a mettere a fattore comune alcune sequenze condivise da più casi d'uso
- **Livello di sommario:** corrisponde ad un obiettivo più ampio che non si ottiene con un solo utilizzo del sistema (es. *negoziare un contratto*)
 - Serve a mostrare il contesto e l'ordine con cui si eseguono più casi a livello obiettivo utente

Diagrammi dei Casi d'Uso


- I diagrammi sono secondari rispetto al testo.
- i casi d'uso sono testo
- Normalmente, i principianti si preoccupano più dei diagrammi che del testo
- I diagrammi devono essere semplici
- Concentrarsi sui casi d'uso, piuttosto che sulle relazioni! 

In conclusione ...

Vantaggi dei Casi d'uso

- Sostituiscono l'elenco dettagliato delle funzioni di basso livello (tipiche di SRS)
- Es.
 - Il sistema accetterà l'inserimento del codice degli articoli
 - Il sistema registrerà i pagamenti con carta di credito nella contabilità...
- I casi d'uso organizzano un insieme di requisiti, raggruppandoli in base agli scenari dei casi d'uso.
- Migliorano la leggibilità e comprensibilità dei requisiti, raggruppandoli

Casi in cui i casi d'uso non sono necessari

- In alcuni sistemi è più opportuno scrivere un elenco di caratteristiche funzionali del sistema
 - Es. acquisizione delle vendite
 - Autorizzazione dei pagamenti 
 - Amministrazione del sistema per gestire utenti, sicurezza...
- Esempi:
 - Un sistema di gestione di una base di dati
 - Un application server
 - Un framework di back-end ...

Casi d'Uso nelle Fasi di UP

Come scrivere i Casi d'Uso durante l'Ideazione in UP

- Non tutti i casi d'uso sono scritti in formato dettagliato in questa fase!
- Si parte dalla tabella Attori- Obiettivi
- Si definisce il diagramma di contesto (Attori e Sistema)
- Poi si elencano in formato breve tutti i principali casi d'uso
- Al termine si passa a dettagliare un primo insieme di casi d'uso più significativi e rischiosi (un 10- 20%)
- Es. *Elabora Vendita e Gestisci Restituzione*

Come scrivere i Casi d'Uso durante l'Elaborazione in UP



- In questa fase si eseguiranno varie Iterazioni
- Durante ogni iterazione sarà svolto un Workshop dei requisiti in cui si sceglieranno e dettaglieranno nuovi casi d'uso
- Alla fine delle iterazioni di Elaborazione, un 80-90% dei Casi d'Uso sarà stato dettagliato
- E alcune parti del sistema saranno anche state implementate

Come scrivere i Casi d'Uso durante la Costruzione


- In questa fase i requisiti si saranno ormai stabilizzati
- Potrebbero però essere ancora descritti alcuni casi d'uso minori e si potranno tenere alcuni Workshop dei Requisiti

Altri elaborati per i Requisiti (Cap. 8 Larman)

Altri elaborati per i requisiti

- Specifiche Supplementari
 - Tutto ciò che non è compreso nei casi d'uso
- Glossario
- Documento di Visione
- Regole di Business

Specifiche Supplementari

- Specifiche FURPS+
- *Funzionalità:*
 - Funzionalità comuni a molti casi d'uso 
 - Es. logging, Gestione degli errori, Sicurezza (autenticazione utente richiesta per ogni utilizzo)
- *Usabilità*
 - Es. Il cliente del POS usa un monitor grande, quindi:
 - Il testo dovrebbe essere visibile a 1 metro di distanza
 - Evitare i colori associati al daltonismo
- *Affidabilità*
 - Ripristinabilità (se si verificano problemi con i sistemi esterni, cercare di completare comunque la vendita)

Specifiche Supplementari

- *Prestazioni*
 - Ottenere risposta dal sistema autorizzazione in meno di 30 sec, nel 90% dei casi
- *Sostenibilità*
 - Adattabilità
 - Configurabilità
- *Vincoli di implementazione*
 - Uso di Java...
- *Componenti acquisitati*
 - Calcolatore imposte
- *Componenti open source...*

Specifiche Supplementari

- Interfacce

- Interfacce hardware significative

- Monitor Touch
 - Lettore Laser di codice a barre
 - Stampante per ricevute
 - Lettore di Carta

- Interfacce software verso sistemi esterni

- Per sistema calcolatore imposte, contabilità, inventario...

- Regole di dominio (di business) specifiche

- R1: regole per lo sconto dei vari tipi di acquirenti
 - R2: regole di sconto su una vendita (> 1000 Euro)
 - R3: Regole di sconto su prodotti particolari

Documento di Visione



- È un documento di sintesi, per comunicare rapidamente le idee più importanti.
- Esempio di struttura:
 - Introduzione (Prevediamo di realizzare un sistema POS...)
 - Parti interessate
 - Utenti
 - Obiettivi delle parti interessate
 - Riepilogo delle caratteristiche del sistema

Glossario dei Dati



- È un elenco di termini significativi e relative definizioni
- In UP il Glossario svolge anche il ruolo di Dizionario dei Dati
- In fase di Ideazione-> solo elenco di termini e descrizioni
- In fase di Elaborazione→ viene esteso con più attributi:
 - Descrizione
 - Formato (tipo, lunghezza, Unità)
 - Relazioni con altri elementi
 - Insieme di valori ammessi
 - Regole di validazione e Sinonimi

Le Storie Utente (Cap. 9 Larman)

Cosa sono le Storie Utente?

- Nei Metodi Agili, i requisiti si definiscono in base a frequenti interazioni con gli utenti, basate su conversazioni. 
- **Una storia Utente descrive, in modo semplice, una funzionalità o un requisito di interesse per un utente del sistema software.** 
- Esempi:
 - Un cassiere può inserire gli articoli acquistati da un cliente
 - Un cassiere può vedere il totale di una vendita
 - Un cliente può pagare in contanti per una vendita
 - Un cassiere può gestire la restituzione di uno o più articoli da parte di un cliente

Storie Utente

- Una storia utente è un modo per rappresentare una funzionalità o un requisito che è di valore per un utente del sistema o per il suo acquirente.
- Prevede tre aspetti:
 - Una **Descrizione scritta** della storia, di alto livello, semplice e breve
 - Una **conversazione sulla storia**, per comprendere i dettagli della storia 
 - Un **insieme di Test di Accettazione** che vengono usati per determinare quando una storia è completa. 

Esempio

- **Descrizione:** Un cassiere può inserire gli articoli acquisitati dal cliente
- La **discussione** tra i membri del team serve a chiarire i dettagli della storia. Es. l'inserimento dell'articolo avviene mediante il codice e si possono inserire più articoli della stessa categoria.
- La discussione produce una serie di **Test di accettazione**, che specificano le aspettative del cliente sulla storia:
 - Provare con un solo articolo
 - Provare con diversi articoli, diversi fra loro
 - Provare con diversi articoli, con più articoli della stessa categoria
 - Provare con tanti articoli

Storie Utente e Schede di descrizione

- In genere le storie utente si scrivono su piccole schede di carta (*es. Post-It*).
- La scheda deve essere piccola per non inserire troppi dettagli.
- Sul retro della scheda si possono riportare i Test di Accettazione

Un Template per le Storie Utente

È tipico usare un template per scrivere Storie Utente.

Ad esempio:

As a < type of user > ,

I want < to perform some task >

so that I can < achieve some goal/benefit/value>.

Esempio

User story title: Customer withdraws cash.

As a customer,

I want to withdraw cash from an ATM

So that I don't have to wait in line at the bank.

Acceptance Criterion 1:

Given that the account is creditworthy

And the card is valid

And the dispenser contains cash,

When the customer requests the cash

Then ensure the account is debited

And ensure cash is dispensed

And ensure the card is returned.

Acceptance Criterion 2:

Given that the account is overdrawn

And the card is valid,

When the customer requests the cash

Then ensure the rejection message is displayed

And ensure cash is not dispensed.

Dimensione delle Storie



- Nello sviluppo Agile è utile avere storie utente che possono essere implementate e verificate da un programmatore, o una coppia di programmatori, in un tempo compreso fra mezza giornata e al più due settimane.
- Questa dimensione è adatta per la pianificazione iterativa.
 - Una storia utente può essere sviluppata nell'arco di una giornata, oppure entro una iterazione di 2 settimane

Esempi di dimensioni

- 1) Un cassiere può gestire una vendita per una cliente.
 - Troppo Grande!
- 2) Un cassiere può inserire gli articoli acquistati da un cliente.
 - Giusta dimensione!
- 3) La ricevuta della vendita riporta l'indirizzo del negozio
 - Troppo Piccola!

Suddividere e Combinare le storie

- Una storia utente troppo grande viene chiamata **Epoepa (Epic)**.
- Una Epic si può dividere in più Storie Utente.
- Es. Un cassiere può gestire la vendita di un cliente→
 - è una **Epic**.
- Può essere scomposta in più Storie Utente:
 - Un cassiere può inserire gli articoli acquistati da un cliente
 - Un cassiere può vedere il totale di una vendita
 - Un cliente può pagare in contanti per una vendita
 - Un cliente può ottenere una ricevuta per una vendita

Le 6 Proprietà INVEST delle Storie

- INVEST= Independent, Negotiable, Valuable, Estimable, Small, Testable
- **Indipendente:** bisogna evitare dipendenze fra le storie, per evitare problemi nella stima, nell'assegnazione di priorità e nella pianificazione.
- **Negoziabile:** i dettagli della storia vanno negoziati durante la discussione, quindi una storia non deve essere troppo dettagliata
- **Di Valore per utenti e acquirenti:** le storie dovrebbero essere di valore per gli utenti, ma talvolta anche per gli acquirenti, non solo per gli sviluppatori

Le 6 Proprietà INVEST delle Storie

- **Stimabile:** Il team di sviluppo dovrebbe essere in grado di stimare il lavoro richiesto per sviluppare ogni storia utente.
- **Piccola:** La dimensione dovrebbe essere tale da permetterne lo sviluppo in un tempo pari a mezza giornata o al più due settimane.
- **Verificabile:** devono essere scritte in modo che siano verificabili, per capire se sono state sviluppate con successo e concluse.

Esempi di Storie per il Gioco che simula una partita al Monopoly

- Creare un tabellone, i dadi, e quanto altro serve per simulare una partita, tra un numero di giocatori variabile fra 2 e 8.
- Consentire ad un giocatore di lanciare i dadi e muovere di conseguenza il segnalino sul tabellone.
- Tutti i giocatori si muovono a turno fino a quando un giocatore non vince la partita, oppure per un numero massimo di giri.
- Visualizzare la traccia di gioco del turno di un giocatore.
- Se un giocatore finisce alla casella del Via, allora riceve 200 dollari.
- Se un giocatore finisce su casella tassa patrimoniale, allora paga la tassa patrimoniale.
- Se un giocatore finisce su casella Vai in prigione, allora va in prigione.
- Se un giocatore finisce su casella che non ha proprietario, allora può acquistare quella proprietà....

Riferimenti

- Interessante Video di **Mike Cohn** su Youtube che illustra molti aspetti delle Storie Utente:
- <https://www.youtube.com/watch?v=6q5-cVeNjCE&t=2138s>
- Mike Cohn è uno dei fondatori della Scrum Alliance , ha iniziato la sua carriera nei primi anni '80 come programmatore in APL e BASIC prima di passare a C++ e Java. È autore di diversi libri su Agile