

Le forme dei requisiti



Corso di Ingegneria del Software
Università di Bologna

Obiettivi della lezione

- Cosa sono i requisiti di un software?
- La forma dei requisiti:
 - Frasi strutturate, scenari, casi d'uso, user stories
- Approfondimento sulle user story

Requisito:

definizione dal vocabolario italiano

- Ciascuna delle qualità necessarie e richieste per uno scopo determinato.
- esempio: *ha tutti i requisiti per diventare un bravo informatico*

I requisiti sono desideri!



Come si esprimono i desideri dal punto di vista dello sviluppo software?

Requisito funzionale:

«Il sistema permetterà di prenotare un taxi e di avere una stima del tempo di attesa»

Scenario:

Cliente: “ho bisogno di un taxi in via Zamboni 33”

Sistema: “Fragola 33 arriverà in 3 minuti”

User story:

*Paolo **vuole** poter chiamare un taxi da una app e ottenere risposta con stima del tempo di arrivo, **per** calcolare se arriverà in tempo al suo appuntamento*

Ma cos'è un desiderio?



Legge di Humphrey

I requisiti di un nuovo prodotto software non saranno chiari finché gli utenti non iniziano a usarlo,

ovvero un problema non è ben compreso finché non si sviluppa una soluzione

Gli utenti: “I’ll Know It When I See It” (IKIWISI)

Perché esistono tante forme per i requisiti?

La forma centrata sull'utente

La forma centrata sullo scenario d'uso

La forma contrattuale

La forma adeguata agli sviluppatori

Analisi dei requisiti (Wikipedia)

L'analisi dei requisiti è un'attività preliminare allo sviluppo di un prodotto software, e serve a definire le funzionalità da offrire, ovvero i *requisiti che devono essere soddisfatti* dal software sviluppato

L'analisi dei requisiti è una fase presente in tutti i modelli di ciclo di vita del software, pur con diverse enfasi e diverse connotazioni

Nel nostro corrisponde alla definizione e modifica del backlog di prodotto

Esempio

- Una biblioteca gestisce prestiti di 100.000 volumi a 5.000 iscritti.
- La biblioteca è dotata di un sistema di catalogazione dei libri.
- I volumi sono catalogati con i metadati bibliografici usuali (autore, titolo, editore, anno, ecc.) e identificati mediante il proprio ISBN ed un contatore di copia.
- Ci sono due tipi d'utente: il bibliotecario e l'iscritto; il primo può aggiornare la base di dati, mentre il secondo può solo consultare i dati dei libri. A tutti gli utenti sarà fornita un'interfaccia Web standard utilizzabile anche da casa.
- Un iscritto chiede alla biblioteca il prestito di uno o più volumi alla volta mediante un Web browser; la biblioteca invia al cliente la lista dei volumi disponibili.
- I libri sono prestati agli iscritti della biblioteca e gli iscritti sono identificati sia da un codice numerico, che dal cognome, nome e data di nascita.
- Il bibliotecario accede mediante password alle operazioni d'aggiornamento, mentre l'iscritto accede liberamente alle operazioni di consultazione
- L'applicazione da progettare deve consentire l'inserimento dei dati delle nuove acquisizioni, l'iscrizione di nuovi utenti, la registrazione dei prestiti, il rientro del libro, il controllo del prestito e la consultazione dei libri disponibili mediante i metadati bibliografici.

Discussione

- Questo testo cosa descrive?
- Sapreste scrivere un software in base ad esso, senza ulteriori interazioni col cliente?



La stesura dei requisiti

- Stabilire **cosa** richiede il cliente ad un sistema software (scopo)
- **senza** definire **come** il sistema verrà costruito (funzioni)

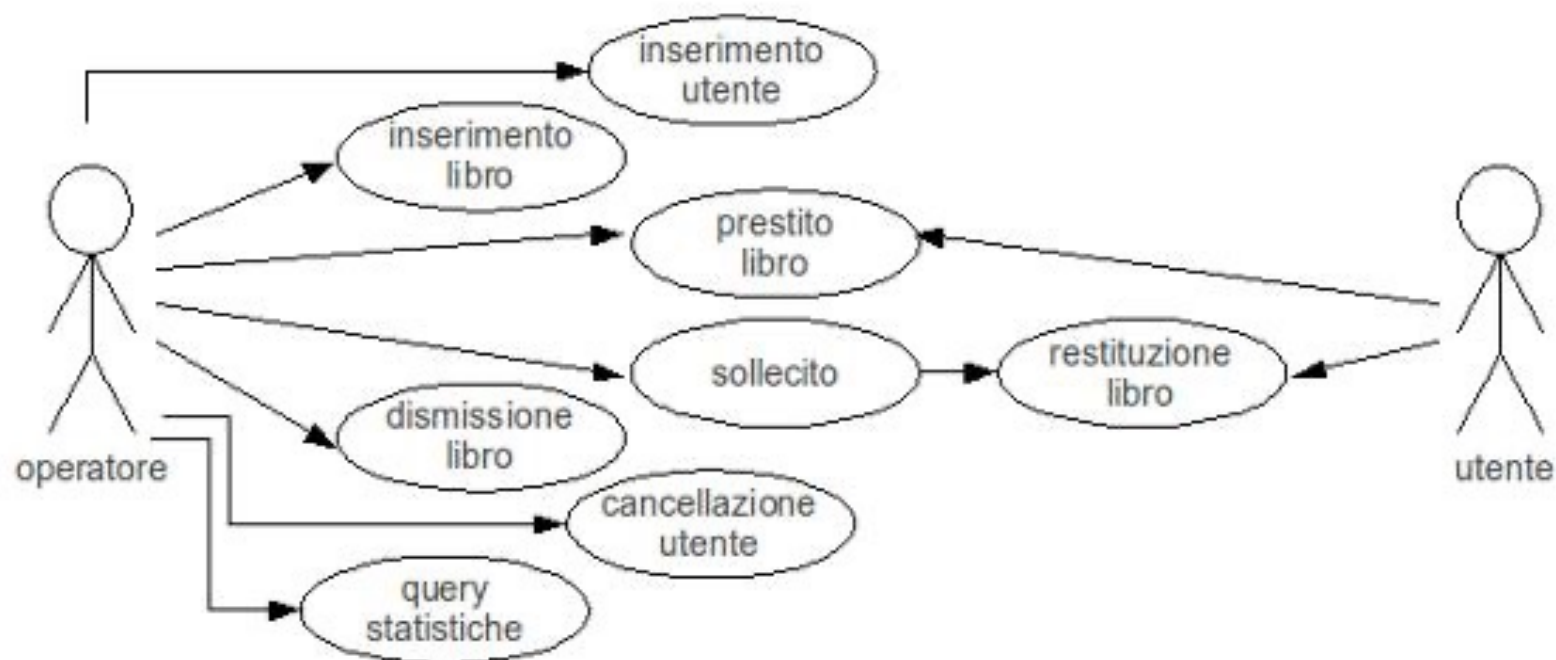
Requisiti per la biblioteca

(scritti in formato funzionale)

1. Il sistema gestirà due tipi di utenti: l'operatore bibliotecario e l'utente «normale»
2. Il sistema permetterà all'utente di ottenere un libro in prestito
3. Il sistema permetterà all'utente di restituire un libro
4. Il sistema permetterà al bibliotecario di inserire un nuovo libro nel catalogo
5. Il sistema permetterà al bibliotecario di dismettere un libro dal catalogo
6. Il sistema permetterà al bibliotecario di inserire un nuovo utente registrato
7. Il sistema permetterà al bibliotecario di cancellare un utente
8. Il sistema permetterà al bibliotecario di approvare un prestito di un libro
9. Il sistema permetterà al bibliotecario di sollecitare la restituzione di un libro
10. Il sistema permetterà al bibliotecario di elaborare statistiche sui prestiti
11. eccetera
12. ...(continua) ...

Requisiti per la biblioteca

(disegnati graficamente come **casi d'uso**)



Esempio di scenario per caso d'uso «prestito»

Titolo del Caso d'Uso: Prestito di un Libro

Attori Principali: Utente

- **Scenario Principale**

1. L'utente avvia l'applicazione della biblioteca.
2. L'utente effettua l'accesso al suo account, fornendo le credenziali di accesso.
3. L'utente cerca il libro desiderato utilizzando la funzione di ricerca all'interno dell'app.
4. L'utente seleziona il libro che desidera dalla lista dei risultati della ricerca.
5. L'applicazione verifica la disponibilità del libro. Se il libro è disponibile, procede col prestito; altrimenti, mostra un messaggio di avviso.
6. L'utente conferma la richiesta di prestito.
7. L'applicazione registra il prestito del libro a nome dell'utente.
8. L'applicazione fornisce una conferma del prestito, includendo la data di scadenza prestito.
9. L'utente può ora visualizzare il libro nel proprio account come libro in prestito.

- **Scenario Alternativo**

- 6a. Se il libro non è disponibile, l'applicazione mostra un messaggio di errore e l'utente può decidere se cercare un altro libro o uscire dall'app.

Requisiti

- I requisiti sono **descrizioni** di ciò che occorre e andrebbe realizzato
- Sono descrizioni di 1) **cosa** il sistema dovrebbe fare, oppure 2) di una **proprietà** o attributo del sistema
- Possono anche essere 3) un **vincolo** sul processo di sviluppo del sistema

Esempio:

- **Cosa** dovrebbe fare il sistema: permettermi di comprare un biglietto del treno
- **Attributo** del sistema: l'acquisto dovrebbe richiedere non più di un minuto
- **Vincolo**: l'acquisto deve avvenire su un telefono Android

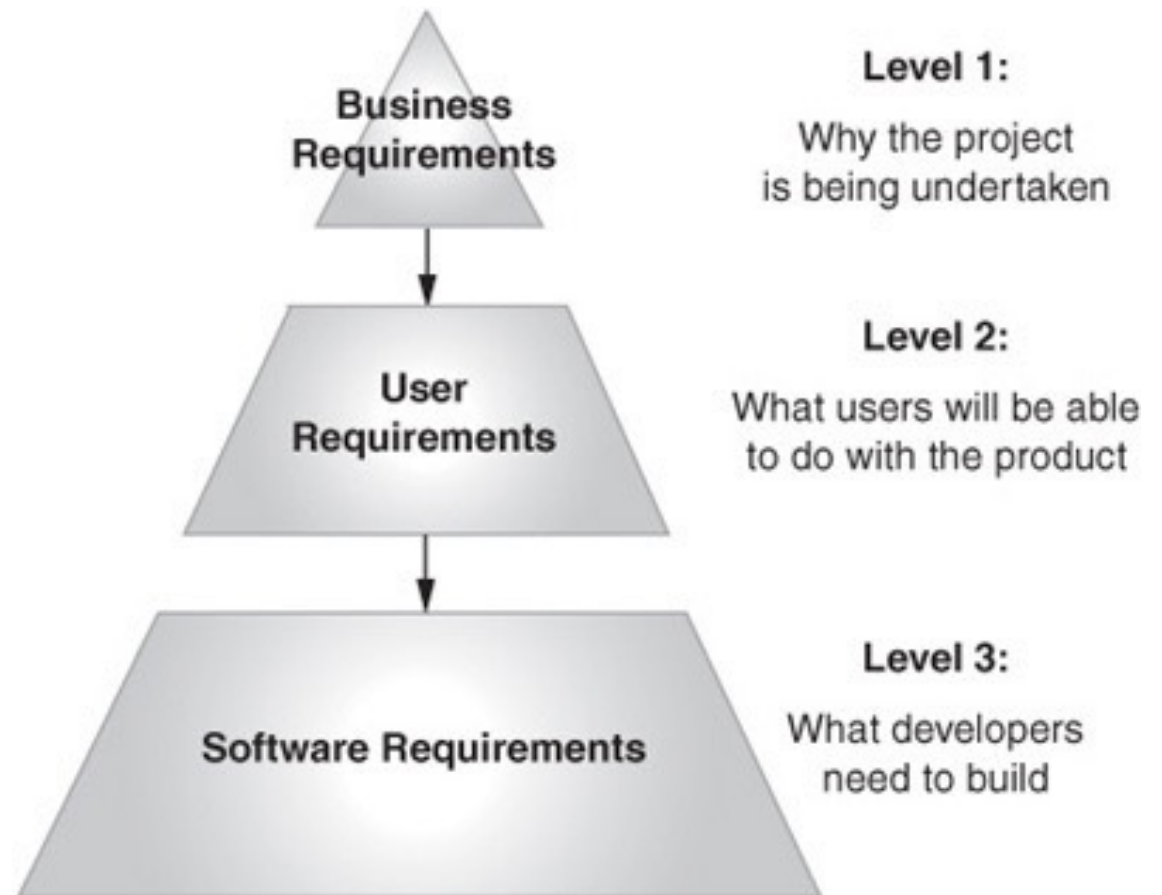
Requisito

Il requisito dice <chi> vuole <cosa> e <perché>

Un requisito dice **cosa** un sistema dovrebbe fare, ma non **come**

I requisiti possono essere:

- di business
- di utente
- del software



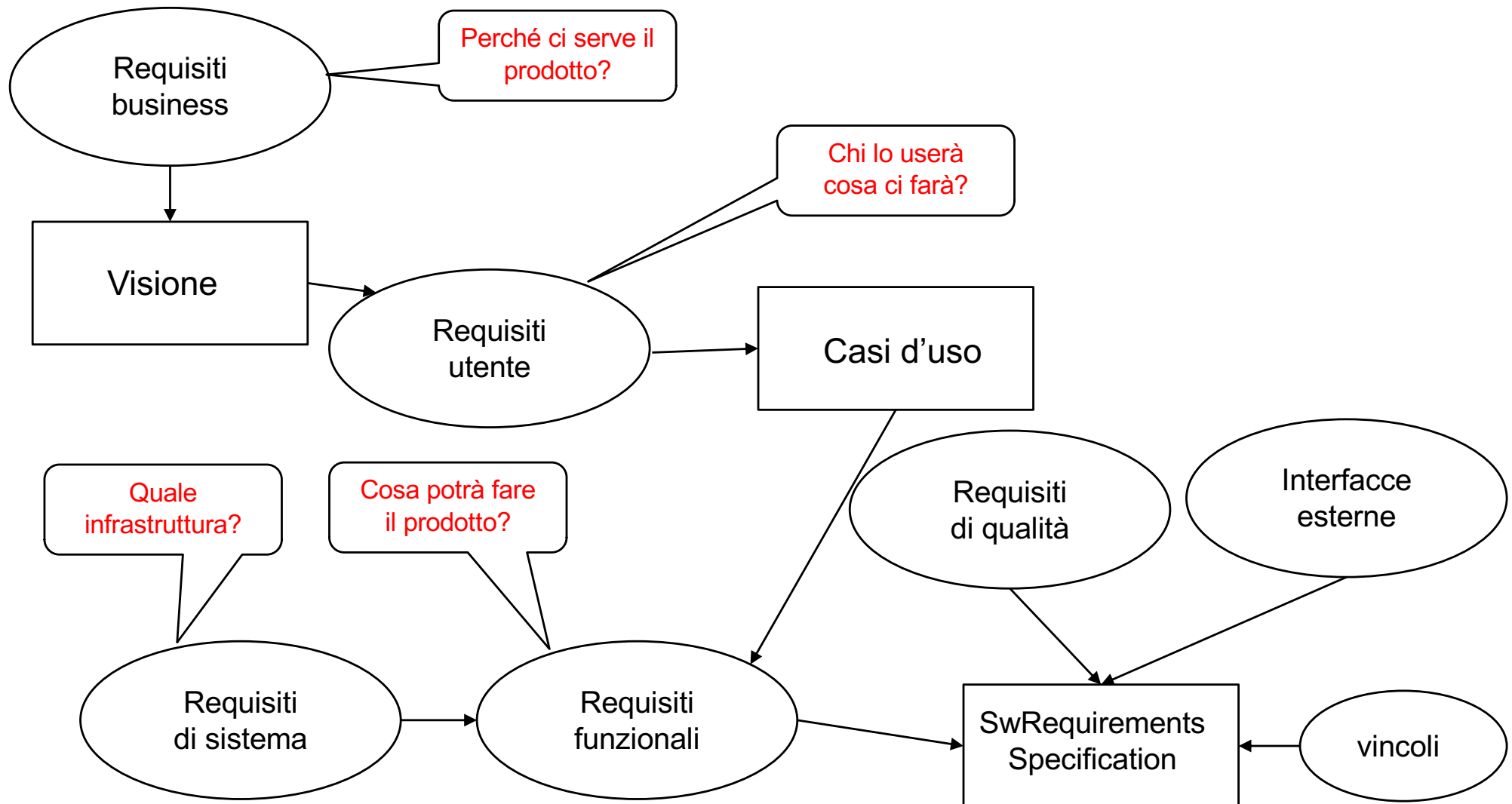
Esempi

Requisito di business: *il sistema biglietteria deve gestire almeno 1M di biglietti al giorno*

Requisito utente: *come viaggiatore voglio comprare un biglietto da Bologna a Milano per far viaggiare mia mamma*

Requisito software: *il record Biglietto conterrà il campo Persona di tipo vettore di stringhe*

Ci sono molti tipi di requisiti e molti modi di specificarli



Discussione

Con quale formato possiamo scrivere i requisiti funzionali di un prodotto digitale?

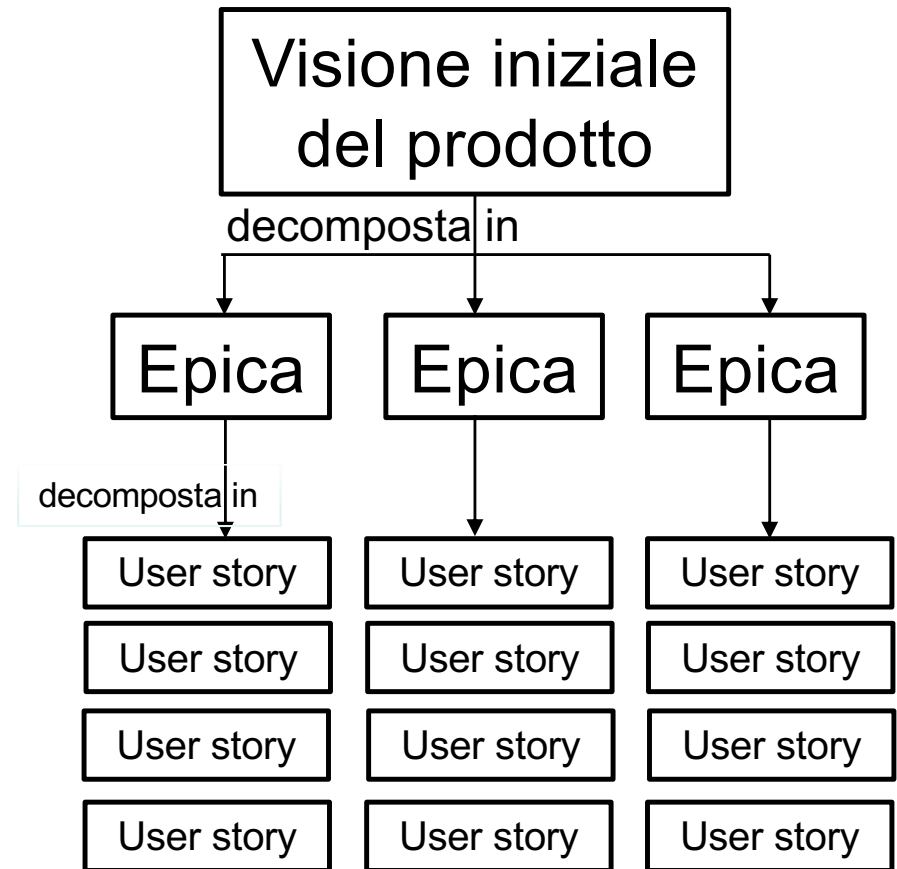


La forma del requisito

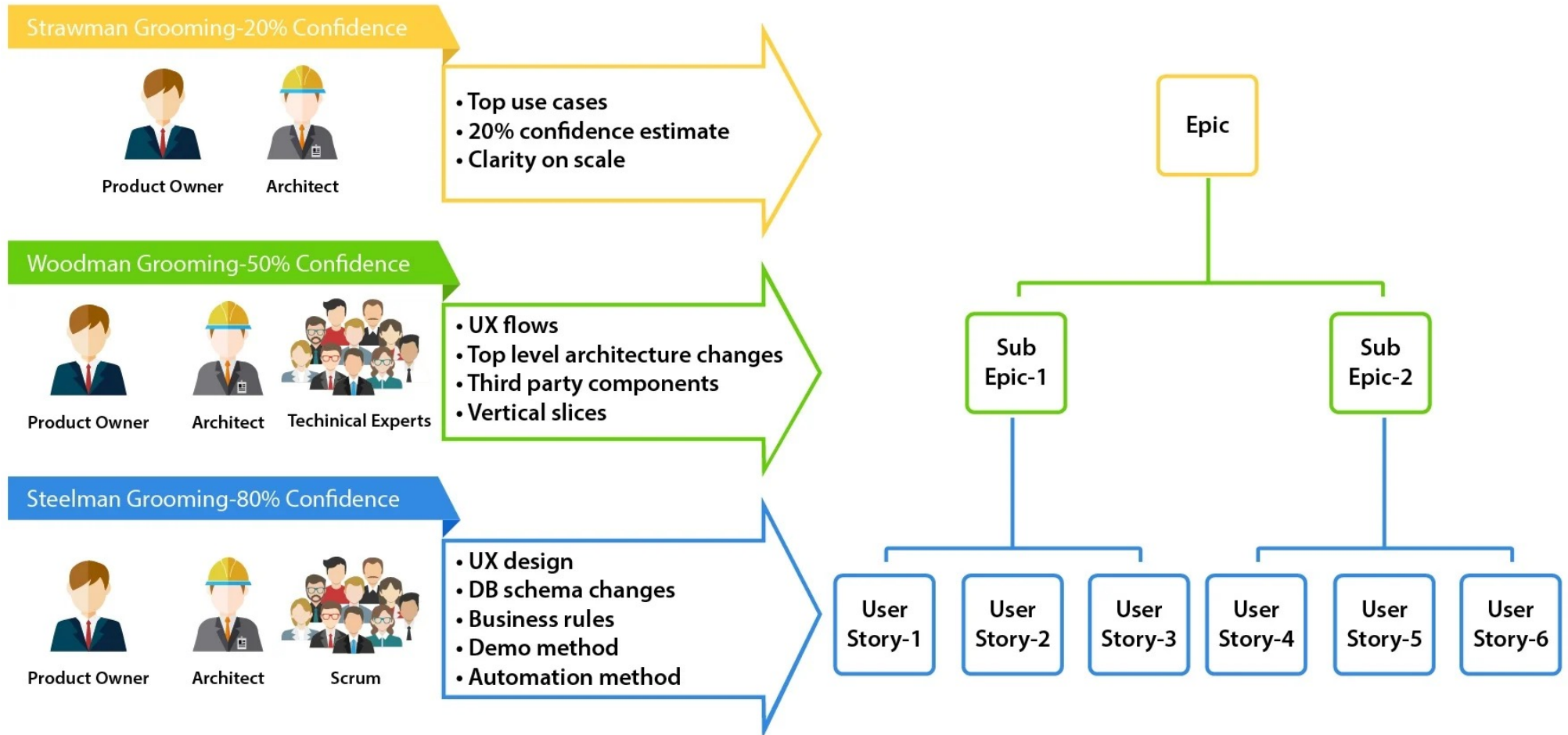
- **Epica:** *«La biblioteca ha bisogno di una app per prestiti e restituzioni dei libri»*
- **User story:** *«Come iscritto voglio consultare il catalogo per prendere in prestito un volume»*
- **Caso d'uso:** *«Un iscritto interroga la funzione Catalogo per cercare un volume, il Catalogo chiede all'iscritto i dati necessari alla ricerca, l'iscritto inserisce i dati, il Catalogo risponde...»*
- **Funzionale:** *«L'applicazione gestirà i prestiti dei volumi agli iscritti»*

Epiche

- L'*epica* rappresenta un insieme di funzionalità ad alto livello richieste in una visione di prodotto
- L'*epica* aiuta a identificare i requisiti principali di un prodotto e fornisce una visione d'insieme di ciò che deve essere sviluppato.
- In genere, un'*epica* è troppo grande per essere implementata in una singola iterazione di sviluppo e deve essere decomposta in più user story
- la user story rappresenta una singola funzionalità implementabile in una sola iterazione



Raffinare il backlog



I requisiti con le user stories

La forma agile dei requisiti: le user stories

Nei modelli di sviluppo agili i requisiti si scrivono mediante le **user stories**, che hanno questa forma:

Come *<tipo di utente>* *(ruolo)*

Voglio *<fare qualcosa>* *(funzione)*

Per *<conseguire un obiettivo>* *(scopo)*

Esempi:

Come *utente della biblioteca*

Voglio *consultare il catalogo*

Per *prendere a prestito un volume*

Come *bibliotecario*

Voglio *consultare il catalogo*

Per *aggiungere un nuovo libro*

1 Individuare gli utenti

- Una user story descrive come qualcuno usa un prodotto
- Se non sapete ancora chi saranno gli utenti oppure come useranno il prodotto NON scrivete le user story

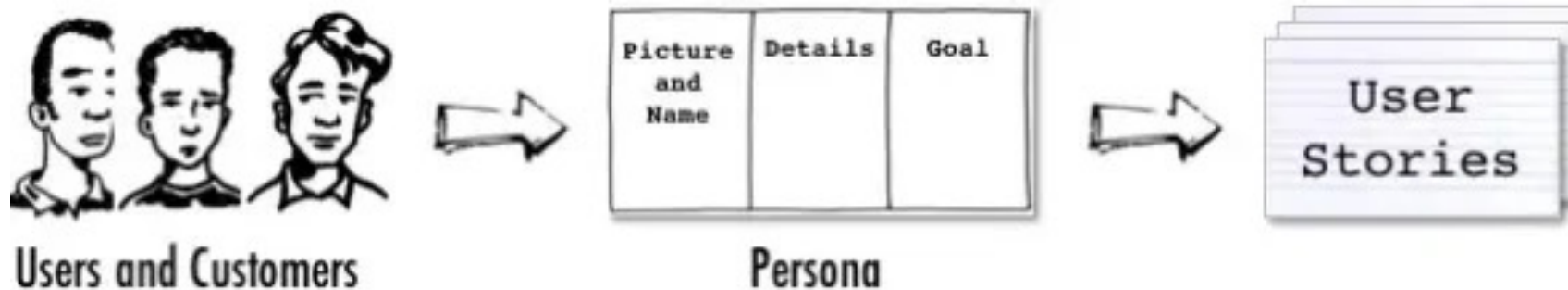


2 usare utenti ideali (*personae*)





Le *personae* sono personaggi immaginari che si basano sulla conoscenza diretta del target del prodotto.


Di solito sono costituiti da: un nome e da un'immagine; alcune caratteristiche e comportamenti rilevanti; e un obiettivo personale.

L'obiettivo è il beneficio che la persona vuole ottenere, o il problema che vuole risolvere usando il prodotto

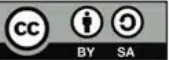


Schema per definire una *persona*

ROMAN'S PERSONA TEMPLATE 		
 PICTURE & NAME	 DETAILS	 GOAL
<p>What does the persona look like? What is its name? Choose a realistic and believable picture and name.</p>	<p>What are the persona's relevant characteristics and behaviours? For instance, demographics, such as age, gender, occupation, and income; psychographics, including lifestyle, social class, and personality; and behavioural attributes like usage patterns, attitudes, and brand loyalty. Only list relevant details.</p>	<p>What problem does the persona want to solve or which benefit does the character seek? Why would the persona want to use or buy the product?</p>



www.romanpichler.com
Template version 04/17

This work is licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License 

3 creare le US collaborando coi devs

Le US sono uno strumento ed un prodotto di conversazioni non solo con gli utenti ma anche con i devs



4 le US sono semplici e concise

Come <persona>
voglio <cosa?>
in modo che <perché>

Nota bene: Questo schema è utile ma non obbligatorio

Nella prossima slide vedremo una forma ammissibile anche se diversa

Forma situazionale di user story

Quando _____ voglio poter _____ cosicché _____
Situazione. Motivazione. Risultato atteso

Quando ho bisogno di un libro voglio poterlo prenotare per averlo al più presto

Quando cambio telefono voglio usare il vecchio per configurare il nuovo

Quando prendo un treno voglio poter comprare il biglietto per partire presto

Requirements

What the software system must do to address the opportunity and satisfy the stakeholders.

Conceived

Bounded

Coherent

Acceptable

Addressed

Fulfilled

Generated by IJ Practice Workbench™
1.1.1

Requirements

Conceived

- ☐ Stakeholders agree system is to be produced
- ☐ Users identified
- ☐ Funding stakeholders identified
- ☐ Opportunity clear

1 / 6

Generated by IJ Practice Workbench™
2018.09

Requirements

Bounded

- ☐ Development stakeholders identified
- ☐ System purpose agreed
- ☐ System success clear
- ☐ Shared solution understanding exists
- ☐ Requirement's format agreed
- ☐ Requirements management in place
- ☐ Prioritization scheme clear
- ☐ Constraints identified & considered
- ☐ Assumptions clear

2 / 6

Generated by IJ Practice Workbench™
2018.09

Requirements

Coherent

- ☐ Requirements shared
- ☐ Requirements' origin clear
- ☐ Rationale clear
- ☐ Conflicts addressed
- ☐ Essential characteristics clear
- ☐ Key usage scenarios explained
- ☐ Priorities clear
- ☐ Impact understood
- ☐ Team knows & agrees on what to deliver

3 / 6

Generated by IJ Practice Workbench™
2018.09

Requirements

Acceptable

- ☐ Acceptable solution described
- ☐ Change under control
- ☐ Value to be realized clear
- ☐ Clear how opportunity addressed
- ☐ Testable

4 / 6

Generated by IJ Practice Workbench™
2018.09

Requirements

Addressed

- ☐ Enough addressed to be acceptable
- ☐ Requirements and system match
- ☐ Value realized clear
- ☐ System worth making operational

5 / 6

Generated by IJ Practice Workbench™
2018.09

Requirements

Fulfilled

- ☐ Stakeholders accept requirements
- ☐ No hindering requirements
- ☐ Requirements fully satisfied

6 / 6

Generated by IJ Practice Workbench™
2018.09

Conclusioni

- Le user story sono una delle modalità preferite per scrivere i requisiti di un prodotto digitale
- La gestione del prodotto arriva a definire le user story dopo un'analisi di mercato che include la caratterizzazione di *personae* ideali utenti del prodotto

Domande di autovalutazione

- Cosa asserisce la legge di Humphrey?
- Cos'è un requisito utente?
- Qual è la forma standard delle user story?
- Cos'è un'epica?
- Quali sono gli stati di un requisito nel modello Essence?

Riferimenti

- Cohn, *User stories applied*, Addison-Wesley, 2004
- Pichler, *Agile product management with Scrum*, Addison-Wesley, 2010
- Sedano, Ralph, Perarire, The Product Backlog, Proc. 41° ICSE, 2019

Domande?

