Scrivere (e leggere) i requisiti di un prodotto software

Prof. Paolo Ciancarini
Corso di Ingegneria del Software
CdL Informatica 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
- Analisi e classificazione dei requisiti
- Strumenti di gestione dei requisiti


I requisiti sono desideri!

Ma cos’è un desiderio?

Desideri e bisogni

…e dammi non quello che io desidero
ma solo ciò di cui ho davvero bisogno.

Signore, insegnami l'arte dei piccoli passi

Antoine de Saint-Exupéry

Esempio

Requisito funzionale:

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

User story:

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

Sistema: “Fragola 33 arriverà in 3 minuti”

Ma cos’è un desiderio? E un bisogno?

Scopo e funzione

- I desideri/bisogni di un cliente possono essere espressi come scopi, ovvero come obiettivi di un business
  - Esempio: voglio viaggiare, mi serve di pianificare un tragitto in treno e comprare il biglietto
- Nel progettare un sistema che permetta di conseguire uno scopo dovremo specificare una serie di funzioni
- La relazione tra scopi e funzioni che li conseguono è l'oggetto dell'analisi dei requisiti

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)

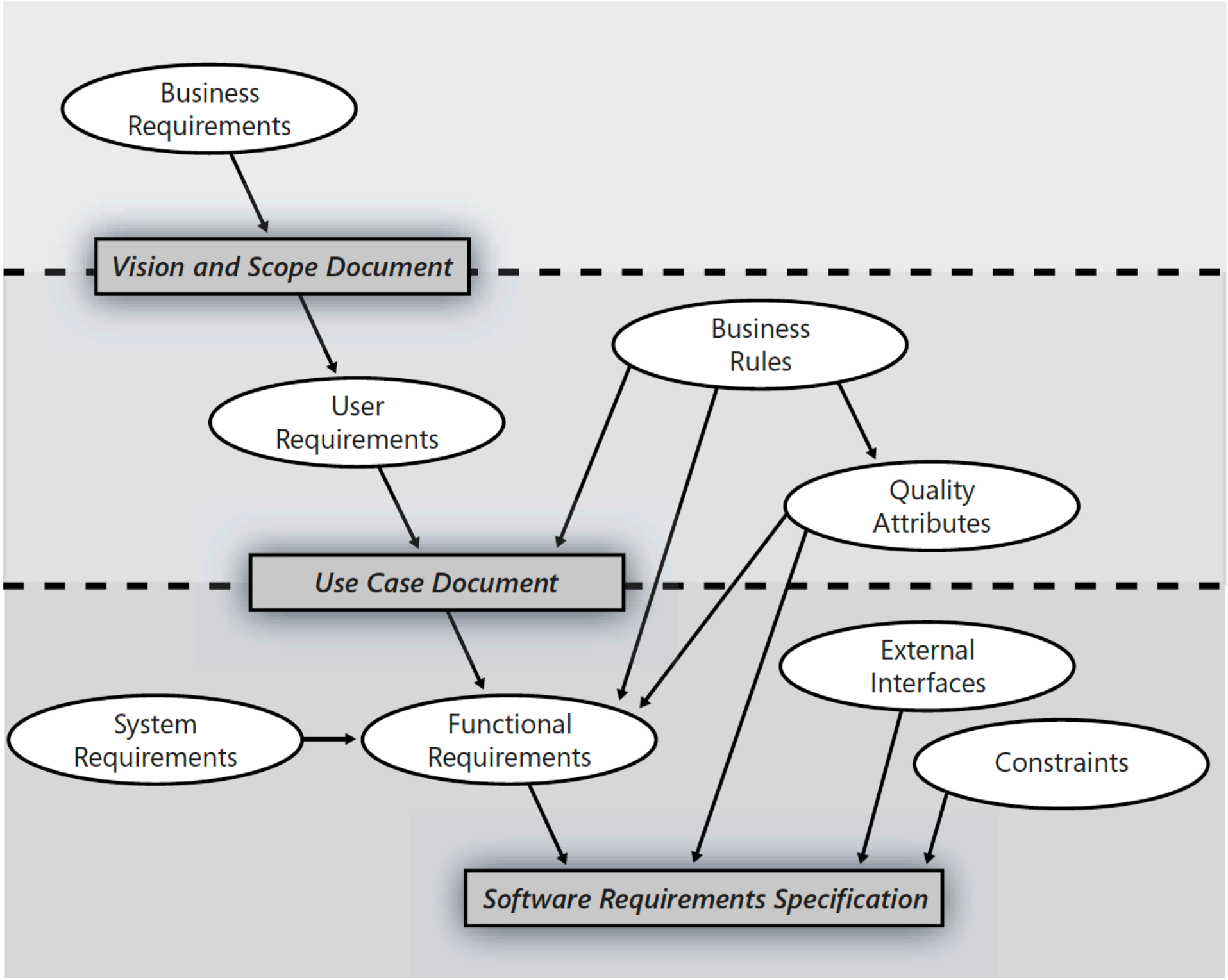
Una descrizione funzionale è un requisito

- F(0) = 1
- F(1) = 1
- F(n) = F(n-1)+F(n-2)

Questi sono requisiti: descrivono un programma che calcola Fibonacci(n)

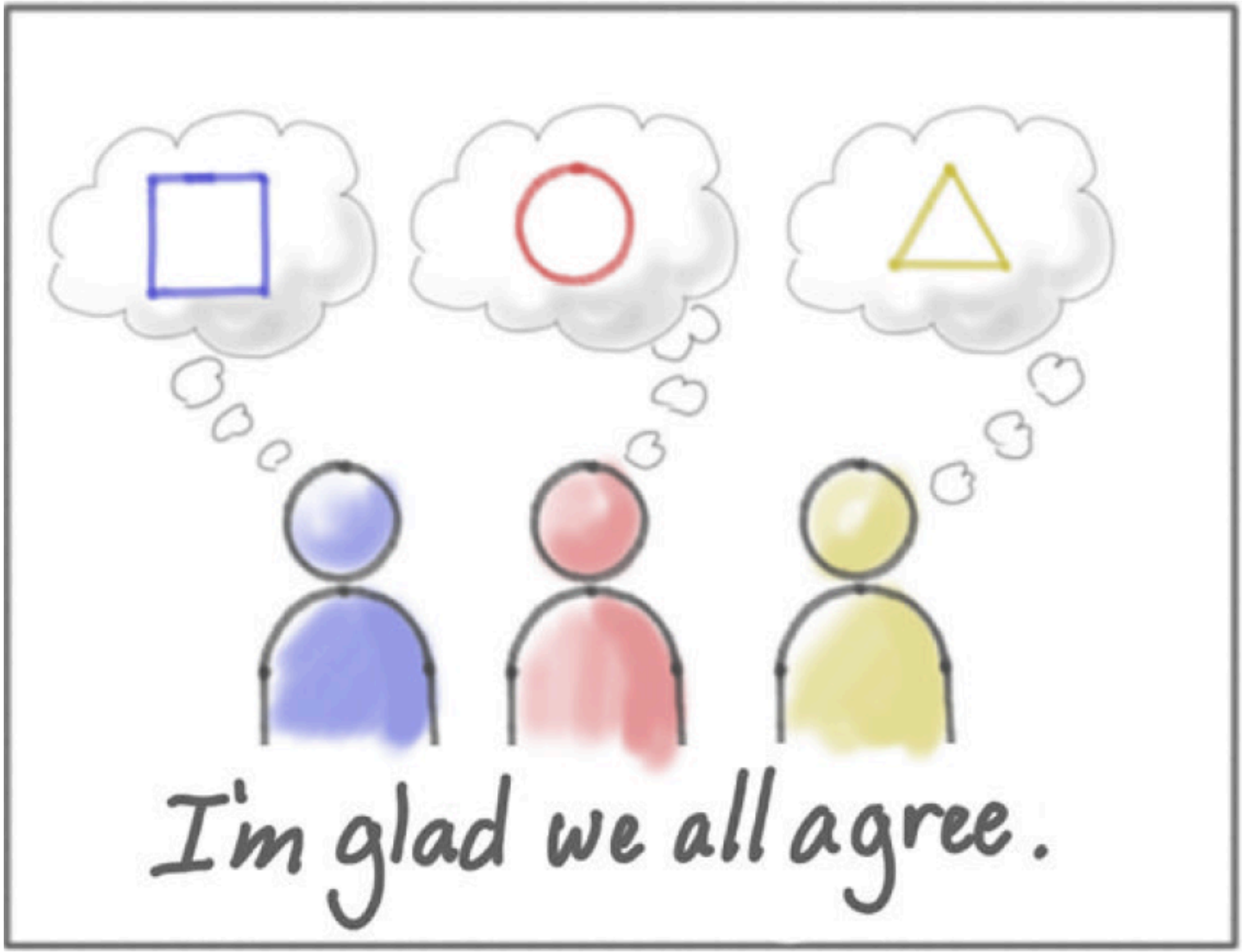
Requisiti

- Requirements are…a specification of what should be implemented.
- They are descriptions of how the system should behave, or of a system property or attribute.
- They may be a constraint on the development process of the system


Business Requirements
Vision and Scope Document
Business Rules
User Requirements
Quality Attributes
Use Case Document
External Interfaces
System Requirements
Functional Requirements
Constraints
Software Requirements Specification

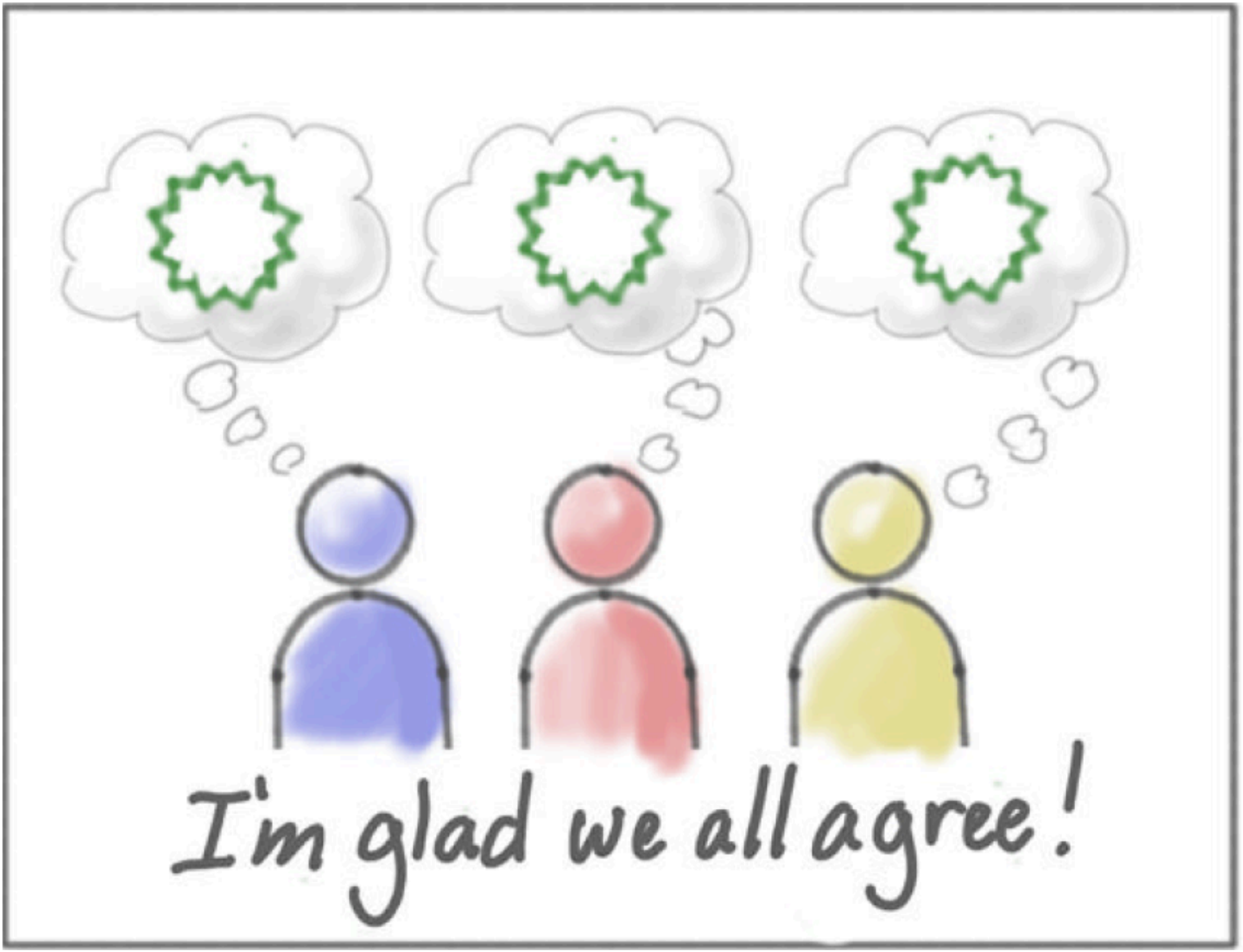
Discussione

Come definiamo i requisiti funzionali di un sistema?


I'm glad we all agree.


oh...


ah ha!


I'm glad we all agree!

La forma del requisito

- **Classica**: *l'applicazione gestirà i prestiti dei volumi agli iscritti*
- **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 le informazioni di ricerca,…*

La risposta agile: le user stories

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

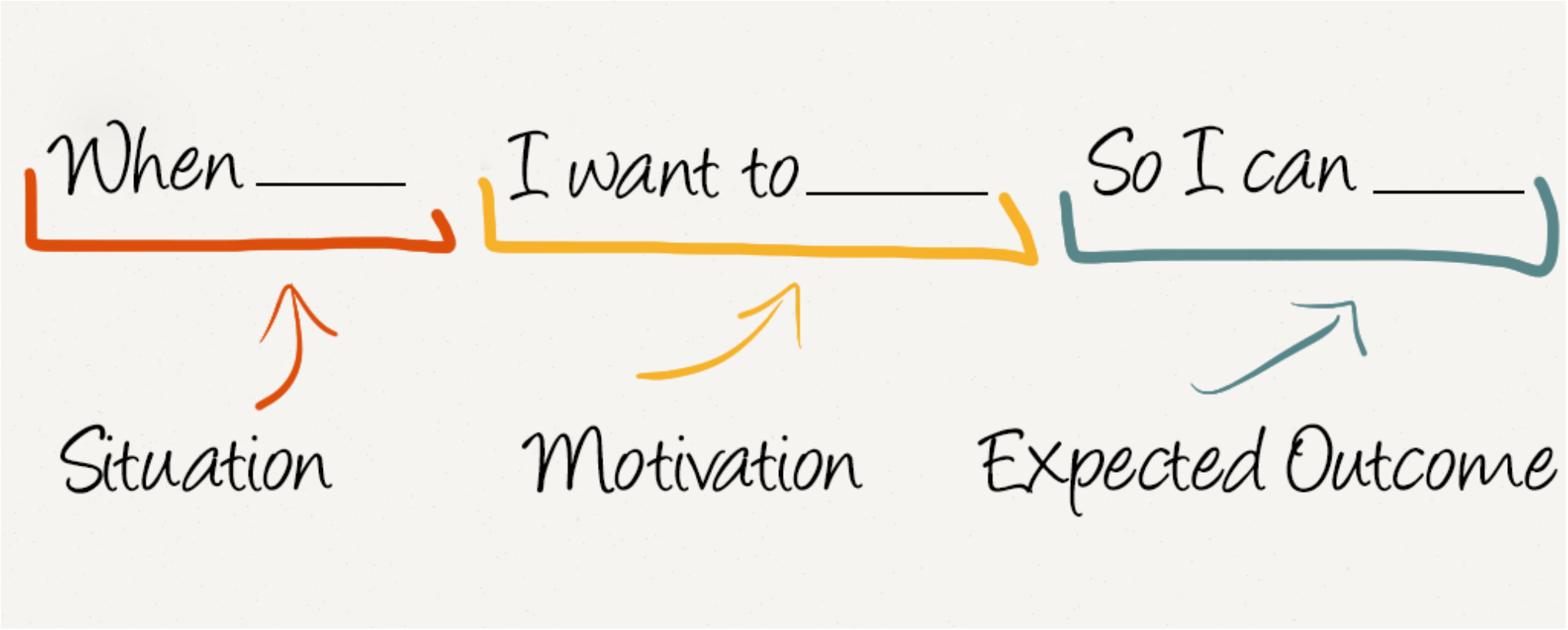
```
As a <user type>
I want <to do something>
so that <I get some value>
```

Come utente della biblioteca (ruolo)

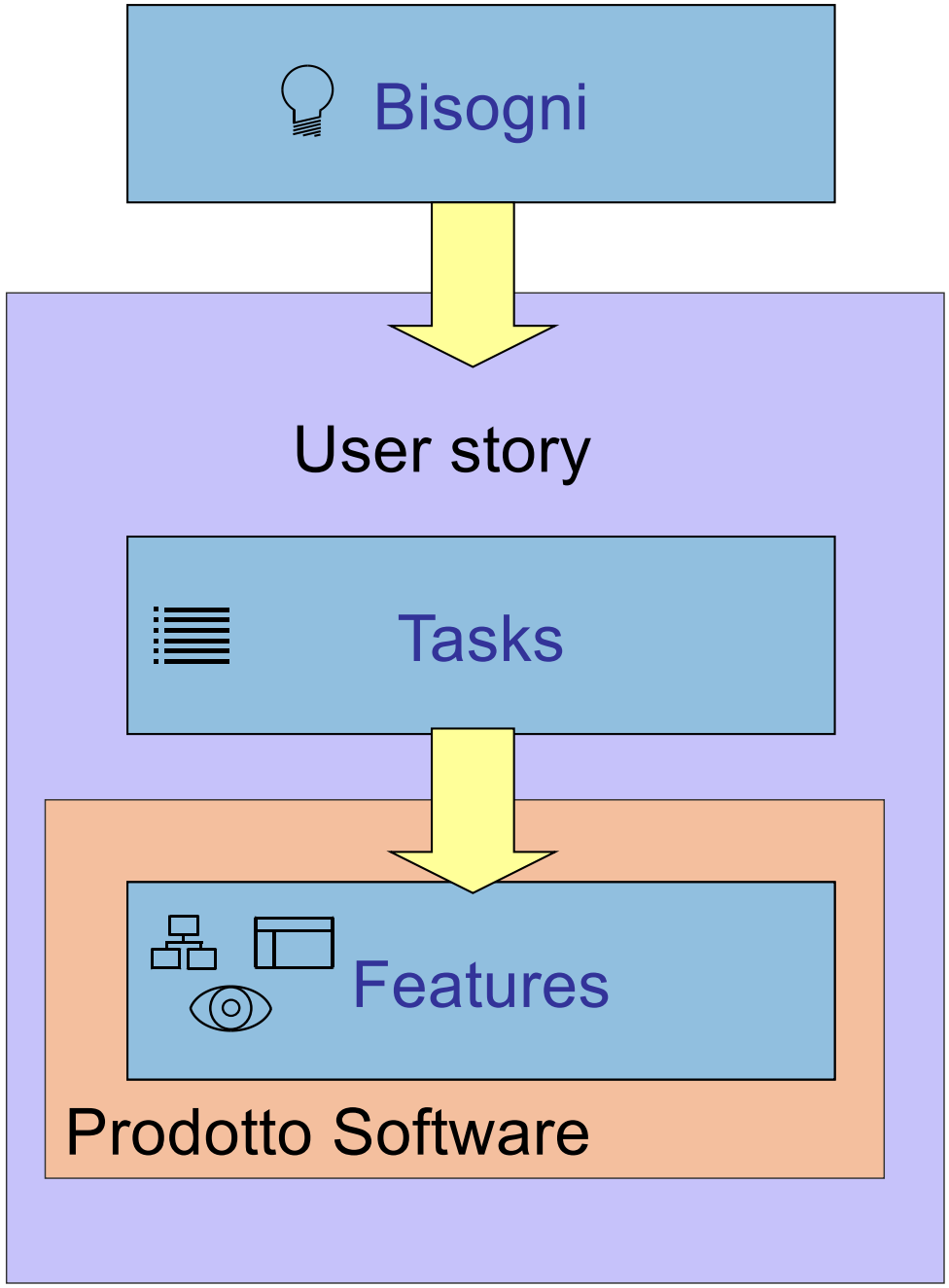
Voglio consultare il catalogo (scopo o funzione)

Per prendere a prestito un volume (valore otttenuto)

Struttura di una user story



Dalle user stories al prodotto



User story: esempio con condizioni

Come utente della biblioteca

voglio consultare il catalogo

per prendere a prestito un volume

Potremmo aggiungere vari dettagli sulle condizioni di svolgimento della user story:

- Considerare se il volume desiderato è già in prestito
- Considerare se l'iscritto ha già in prestito troppi volumi
- Considerare se il volume desiderato è ordinato ma non arrivato
- Considerare se il volume desiderato andrebbe acquistato

User story: esempio con test

- Come utente del portale voglio iscrivermi inserendo il mio CV e lo stipendio desiderato per ricevere informazioni su ogni offerta di lavoro che soddisfi la mia richiesta

- Nota: mostrare stipendio, descrizione, e luogo

- Testare con descrizione CV vuota
- Testare con descrizione CV molto lunga
- Testare con stipendio mancante
- Testare con stipendio a sei cifre

User stories difettose: esempi

ID	Description	Issues
US_1	As a User, I'm able to click a particular location from the map and thereby perform a search of landmarks associated with that latitude longitude combination	Not atomic: two stories in one
US_2	As a care professional I want to see the registered hours of this week (split into products and activities). See: Mockup from Alice NOTE: - First create the overview screen - Then add validations	Not minimal, due to additional note about the mockup
US_3	Add static pages controller to application and define static pages	Missing role
US_4	As a User, I want to open the interactive map, so that I can see the location of landmarks	Conceptual issue: the end is in fact a reference to another story
US_5 US_6	As a User, I'm able to edit any landmark As a User, I'm able to delete a landmark which I added	Conflict: US_5 refers to any landmark, while US_6 only to those that user has added
US_7	As a care professional I want to save a reimbursement. - Add save button on top right (never greyed out)	Hints at the solution
US_8	As a User, I am able to edit the content that I added to a person's profile page	Unclear: what is content here?
US_9	As an Administrator, I am able to view content that needs to be reviewed	The type of content is not specified
US_{10}	Server configuration	In addition to being syntactically incorrect, this is not even a full sentence
US_{11} US_{12}	As an Administrator, I am able to add a new person to the database followed by As a Visitor, I am able to view a person's profile	Viewing relies on first adding a person to the database
US_{13}	As a care professional I want tosee my route list for next/future days, so thatI can prepare myself (for example I can see at what time I should start traveling)	Difficult to estimate because it is unclear what see my route list implies
US_{14}	As an Administrator, I receive an email notification when a new user is registered	Deviates from the template, no "wish" in the means
EP_A US_{15}	As a Visitor, I'm able to see a list of news items, so that I can stay up to date on news As a Visitor, I'm able to see a list of news items, so that I can stay up to date on news	The same equirement is repeated both in epic EP_A, and in a user story US_{14}

A cosa servono le user stories

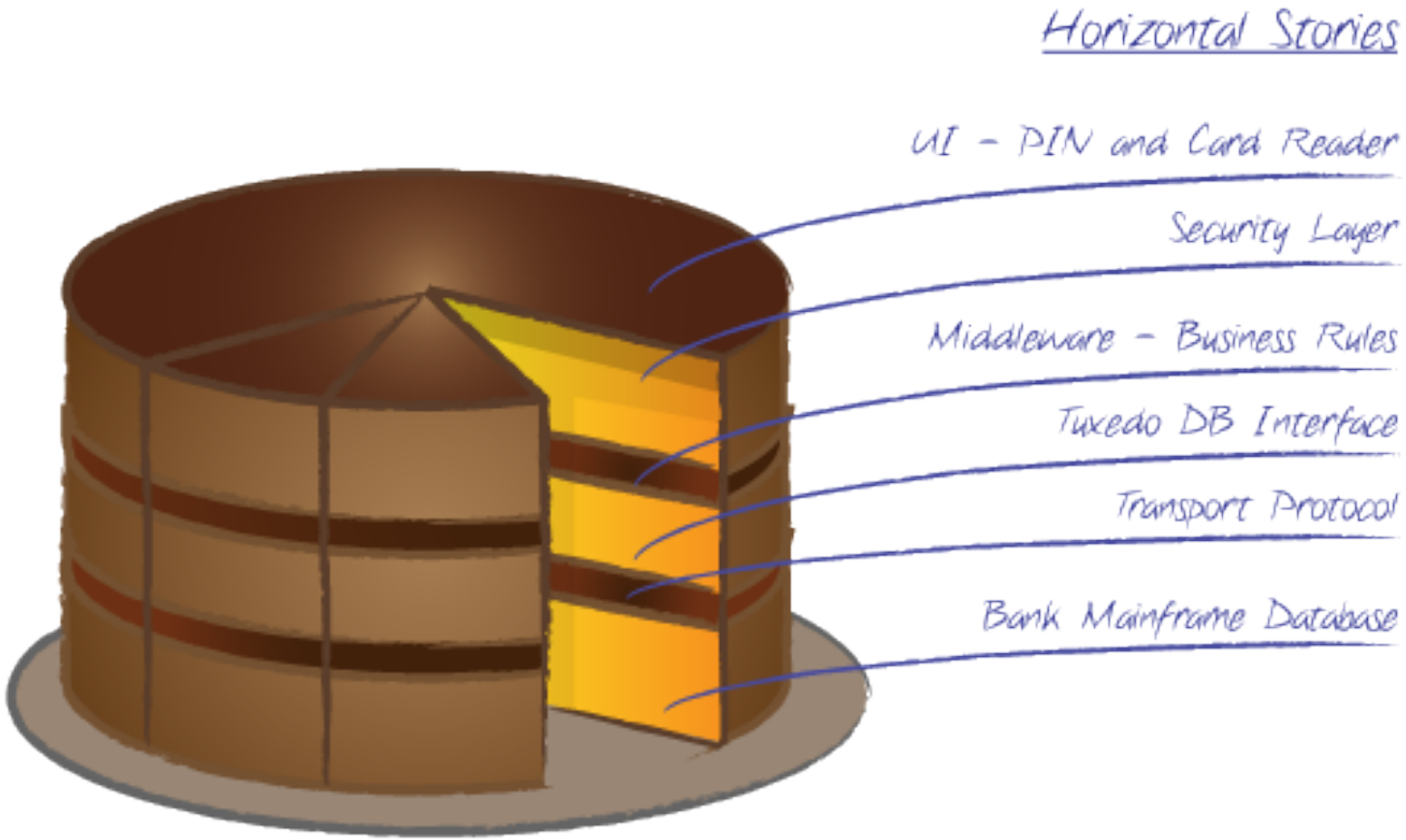
- Il Product Owner scrive le storie e la loro “Definition of Done”, cioè il criterio di accettazione
- Le storie debbono essere sufficientemente piccole da poter essere realizzate in uno o due settimane
- Il team di sviluppo sceglie una storia da realizzare e prepara tutto il necessario perché il PO possa accettare il software risultante, così come richiesto dalla Definition of Done

Non è facile scrivere user stories

- È molto più facile partizionare un backlog *orizzontalmente,* stratificando l'architettura funzionale da costruire
- Gli sviluppatori spesso preferiscono lavorare così per ragioni di efficienza

http://simpleprogrammer.com/2011/11/21/understanding-the-vertical-slice/

Automated Teller Machine (ATM)
Horizontal and Vertical User Stories – Slicing the Cake

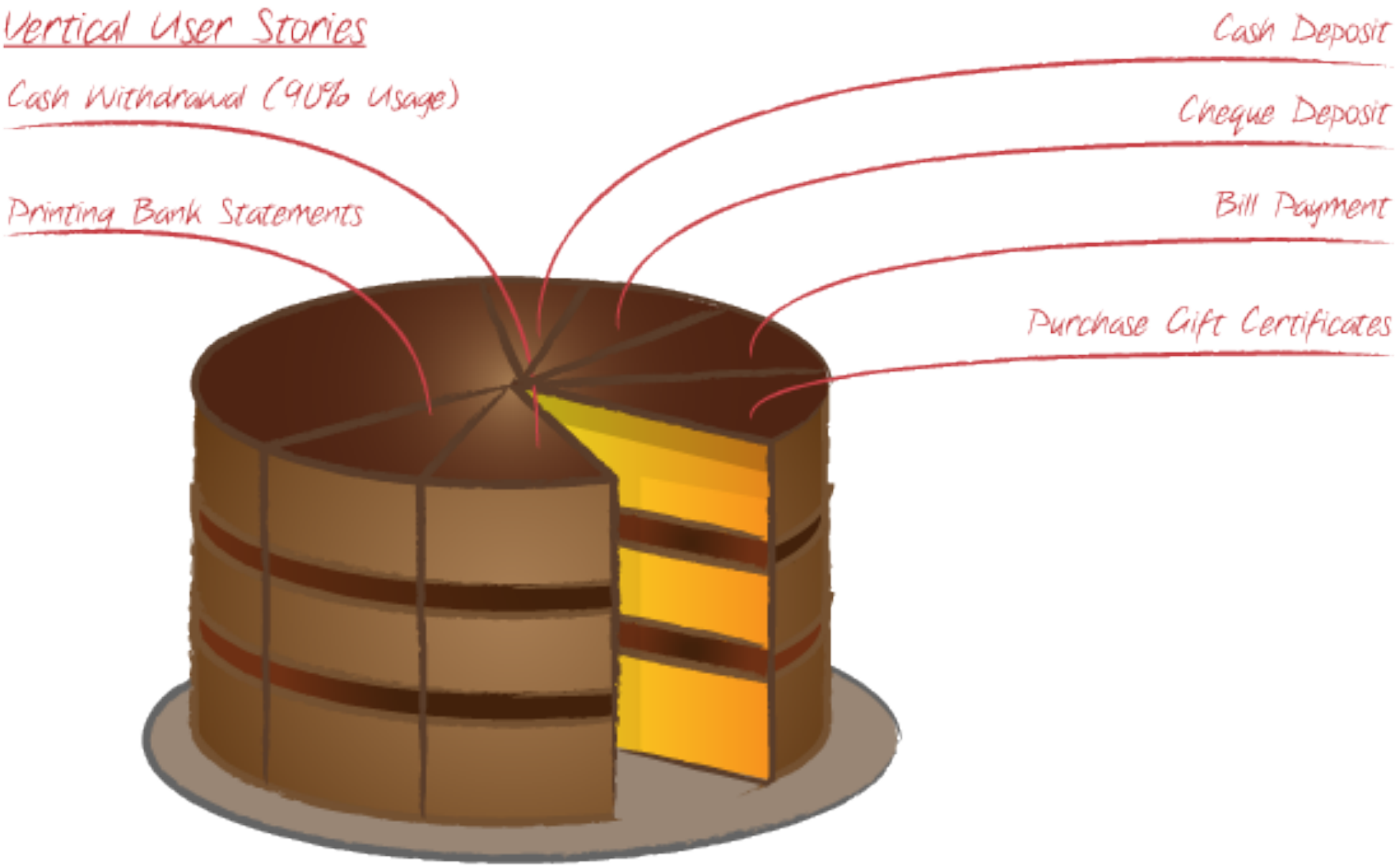


http://www.deltamatrix.com/2012-04-17-04-37-50/horizontal-and-vertical-user-stories-slicing-the-cake

Le partizioni verticali

- Permettono di confezionare e consegnare rapidamente singole funzionalità
- Feedback continuo dal cliente

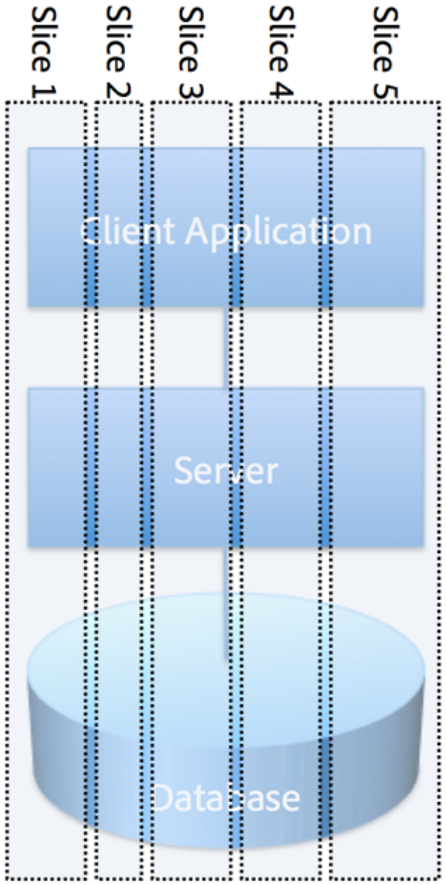
Automated Teller Machine (ATM)
Horizontal and Vertical User Stories – Slicing the Cake



http://www.deltamatrix.com/2012-04-17-04-37-50/horizontal-and-vertical-user-stories-slicing-the-cake

Tecniche per fette verticali

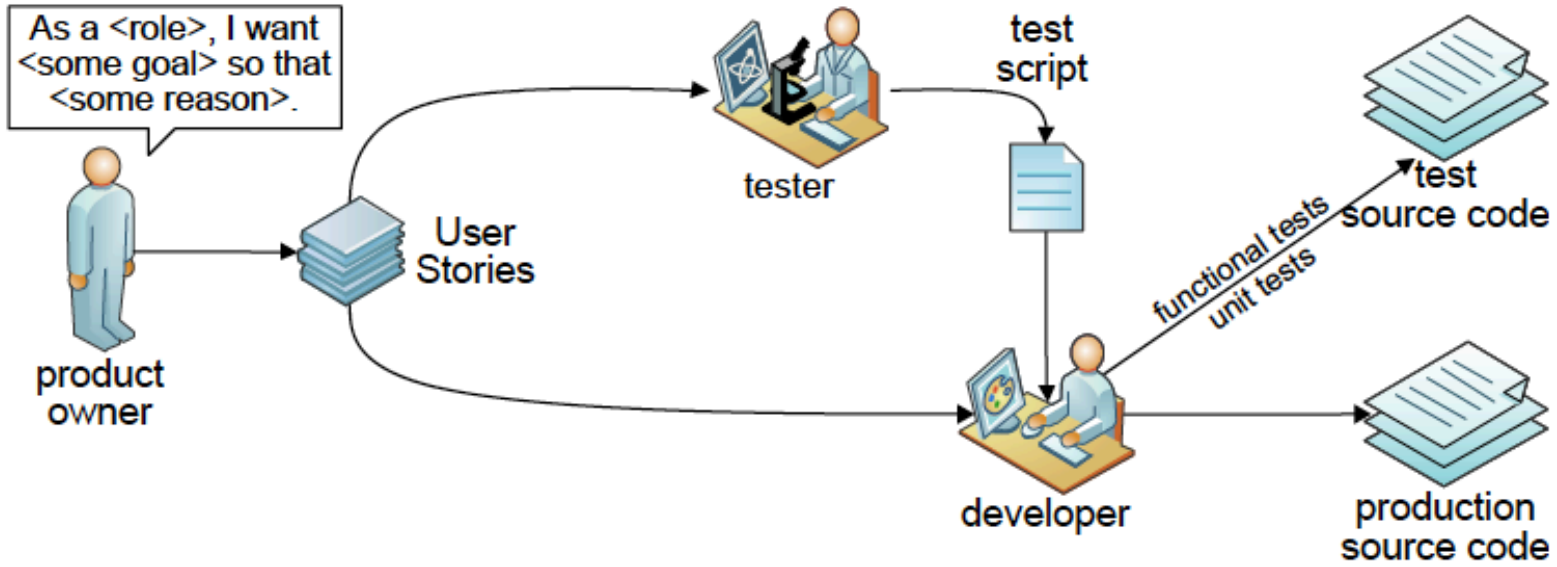
- **Ambiguità** La disambiguazione di termini vaghi può aiutare l'articolazione delle storie
- **Congiunzioni** Una funzione descritta mediante una congiunzione può facilmente essere decomposta
- **Accettazione** Il test di accettazione può indurre ad affettare una storia così da renderla più facilmente testabile
- **Segmentazione** Un compito complesso viene spezzato in più passi semplici



http://blogs.adobe.com/agile/2013/09/27/splitting-stories-into-small-vertical-slices/

User story nel processo agile

- As a <user_type>
- I want to <function>
- so that I can <business value>
- (acceptance criteria of the business value)



Le user story sono frammenti di una conversazione

- Utente: come descrivo i miei desideri?
- Stakeholder: come posso far sì che il prodotto abbia successo?
- PM: come traccio e pianifico questo compito?
- BA: quali sono i dettagli di questa feature?
- UX: quali sono i bisogni dell'utente?
- Developer: quali task devo eseguire oggi?
- QA: come posso validare questo task completato?

I requisiti con UML: i casi d'uso

- Proposti da Ivar Jacobson nel 1992 per il metodo Objectory e ripresi nel RUP
- La ricerca dei casi d'uso (use cases) è *lo studio degli scenari operativi degli utenti di un sistema*
- Gli scenari sono i "modi" in cui il sistema può essere utilizzato (cioè definiscono le operazioni o funzioni che il sistema mette a disposizione dei suoi utilizzatori)

Ivar Jacobson

Scenari

Un caso d'uso si descrive sotto forma di **scenario di interazione** (cioè un dialogo) tra un utente e il sistema

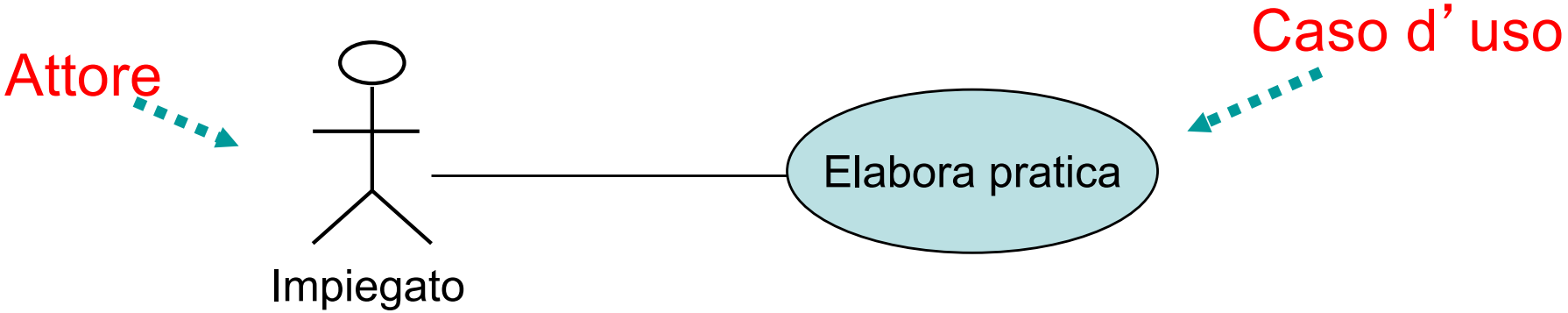
Esempio:

- il cliente richiede l'elenco dei prodotti
- il sistema propone i prodotti disponibili
- il cliente sceglie i prodotti che desidera
- il sistema fornisce il costo totale dei prodotti selezionati
- il cliente conferma l'ordine
- il sistema comunica l'accettazione dell'ordine

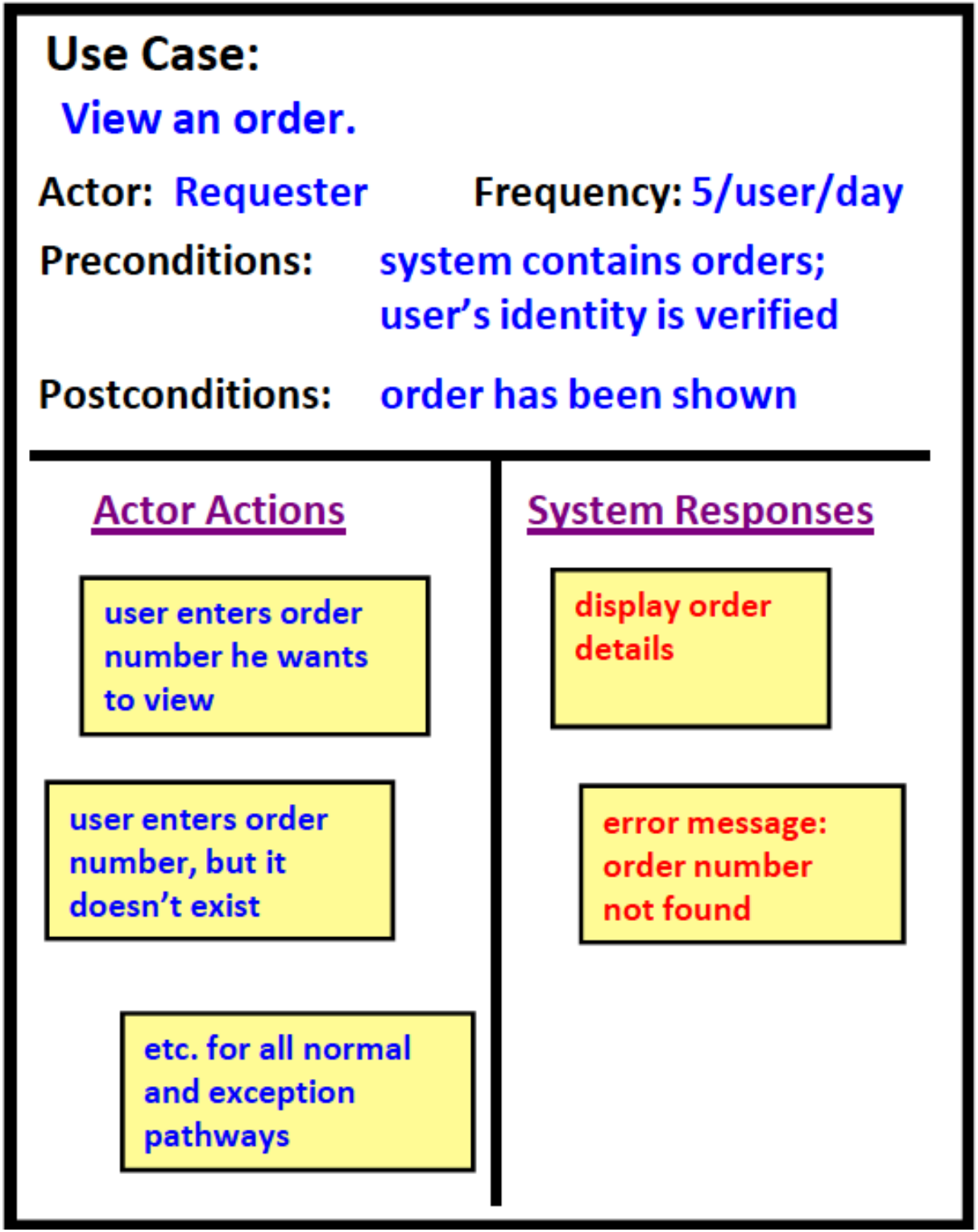
- L'attenzione si focalizza sull'interazione, non sulle attività interne al sistema

Rappresentazione grafica

- Il caso d'uso si rappresenta con un attore che usa una funzione del sistema



Scenario associato ad un caso d'uso



Formato di uno scenario

Use Case ID:			
Use Case Name:			
Created By:		**Last Updated By:**	
Date Created:		**Date Last Updated:**	

Actors:	
Description:	
Trigger:	
Preconditions:	
Postconditions:	
Normal Flow:	
Alternative Flows:	
Exceptions:	
Includes:	
Priority:	
Frequency of Use:	
Business Rules:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Gherkin

Gherkin è un linguaggio testuale parte di Cucumber, uno strumento di test.

Gherkin permette di descrivere test su casi d'uso senza dover dare dettagli sull'implementazione. Gherkin usa l'indentazione

```
Feature: Serve coffee
        Coffee should not be served until paid for
        Coffee should not be served until the button has been pressed
        If there is no coffee left then money should be refunded
Scenario: Buy last coffee
        Given there are 1 coffees left in the machine
           And I have deposited 1$
        When I press the coffee button
        Then I should be served a coffee
```

https://github.com/cucumber/cucumber/wiki/Gherkin

Gherkin

```
1: Feature: Some terse yet descriptive text of what is desired
2:   Textual description of the business value of this feature
3:   Business rules that govern the scope of the feature
4:   Any additional information that will make the feature easier
to understand
5:
6: Scenario: Some determinable business situation
7:   Given some precondition
8:    And some other precondition
9:   When some action by the actor
10:   And some other action
11:   And yet another action
12:  Then some testable outcome is achieved
13:   And something else we can check happens too
14:
15: Scenario: A different situation
16: ...
```

github.com/cucumber/cucumber/wiki/Gherkin

Behavior driven development (BDD)

Story: Returns go to stock

As a store owner

In order to keep track of stock

I want to add items back to stock when they're returned.

Scenario 1: Refunded items should be returned to stock

Given that a customer previously bought a black sweater from me **And** I have three black sweaters in stock.

When he returns the black sweater for a refund

Then I should have four black sweaters in stock.

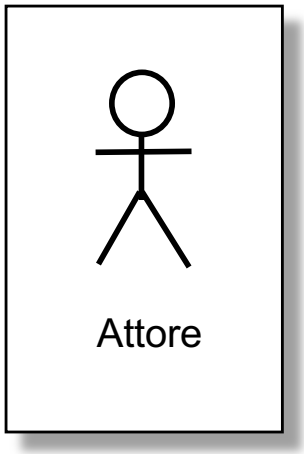
Scenario 2: Replaced items should be returned to stock

Given that a customer previously bought a blue garment from me **And** I have two blue garments in stock **And** three black garments in stock.

When he returns the blue garment for a replacement in black

Then I should have three blue garments in stock **And** two black garments in stock.

Attore



- **Ruolo** che qualcuno o qualcosa ricopre rispetto al sistema
- Gli attori sono esterni al sistema
- Gli attori eseguono casi d'uso
  - Prima si cercano gli attori, poi i loro casi d'uso
- Gli attori "ottengono valore" dal caso d'uso o vi partecipano
- Gli attori possono NON essere persone (es.: altri sistemi, dispositivi hardware)!

Come trovare gli attori

- Chi o cosa è interessato al sistema?
- Chi o cosa modificherà i dati del sistema?
- Chi o cosa vuole informazioni dal sistema?
- Chi o cosa si interfaccerà col sistema?

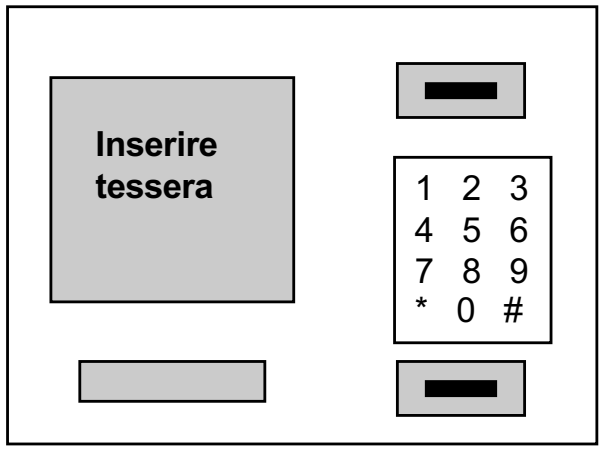
Ricerca degli attori: domande utili

- Chi è interessato a questo requisito?
- Chi utilizzerà questa funzione?
- Quali attori sono in relazione con gli Use Cases?
- Un attore ha più ruoli? Lo stesso ruolo è assegnato a più attori?

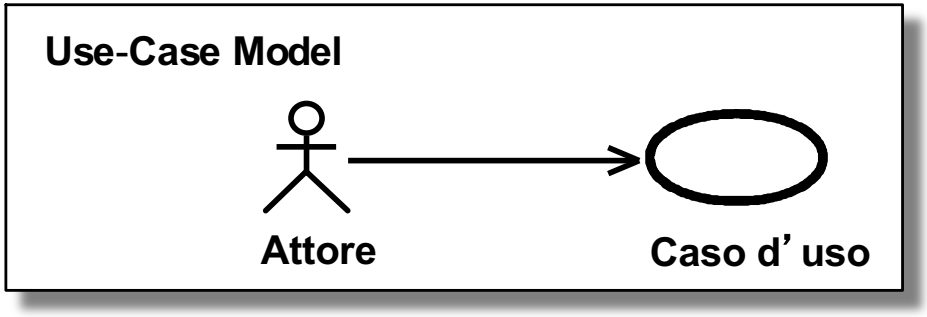
- In quale parte dell'organizzazione è utilizzato il sistema?
- Chi fornirà, utilizzerà ed eliminerà informazioni dal sistema?
- Chi supporterà e manterrà il sistema?
- Il sistema utilizzerà una risorsa esterna?

Istanze di attori

Paolo agisce come attore



Laura agisce come attore



Caso d'uso

- Rappresenta un **requisito funzionale**
- Esplicita cosa ci si aspetta da un sistema ("what?")
- Nasconde il comportamento del sistema ("how?")
- E' una sequenza di **azioni** (con varianti) che producono un risultato osservabile da un **attore**

Si usa per

- Descrivere requisiti d'utente (analisi iniziale)
- Controllare il sistema (testing e verifica)

Caso d'uso

- Ogni sequenza (detta *scenario*) rappresenta l'interazione di entità esterne al sistema (*attori*) con il sistema stesso o sue componenti
- Separa il *flusso principale* dalle varianti *alternative*

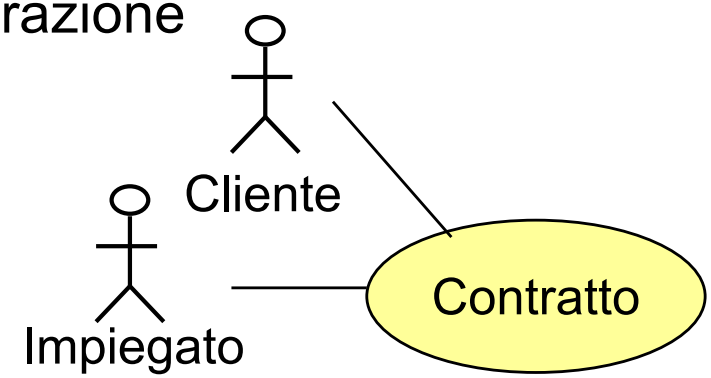
Esempio

Nome: Contratto di acquisto di azioni

Precondizione: l'impiegato è connesso

Flusso principale degli eventi:

1. Inserire nome cliente e numero conto
2. Controllare la loro validità
3. Inserire il numero di azioni da comprare e ID azienda quotata
4. Determinare il prezzo corrente delle azioni
5. Controllare il saldo del cliente
6. Mandare l'ordine alla Borsa
7. Memorizzare numero di conferma dell'operazione



Descrizione dello scenario

- E’ una descrizione generica e sequenziale del flusso di eventi di un caso d’uso
  - Descrivere la precondizione (stato iniziale del sistema)
  - Elencare la sequenza di passi
- Include le interazioni con gli attori e descrive quali entità vengono scambiate
- La descrizione dev’essere chiara, precisa e breve

Ricerca dei casi d’uso: domande utili

- Quali sono i compiti di questo attore?
- L’attore gestirà le informazioni del sistema?
- Quali Casi d'Uso creeranno, modificheranno, leggeranno questa informazione?
- L’attore deve informare il sistema di cambiamenti improvvisi?
- L’attore dev’essere informato di certe situazioni?
- Il sistema supporta il business con un comportamento corretto?
- Quali Casi d'Uso supportano e manutengono il sistema?
- I requisiti funzionali sono tutti coperti dai Casi d'Uso che abbiamo trovato?

Fonti di informazione per i casi d'uso

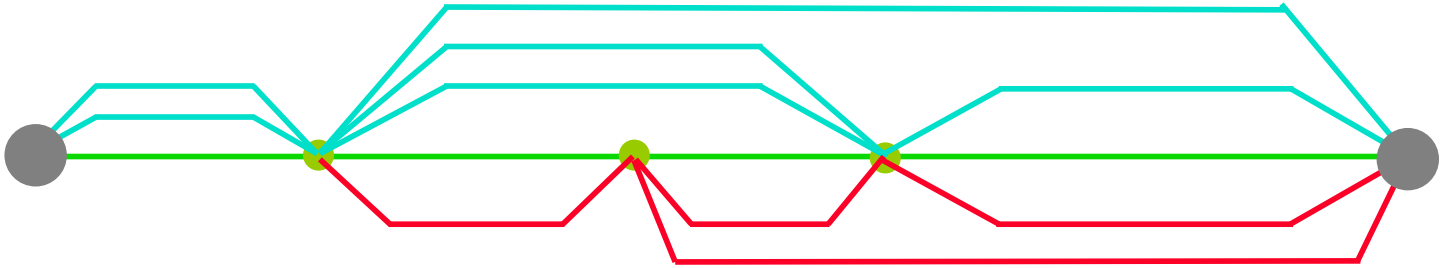
- Documenti di specifica del sistema
- Bibliografia del dominio del sistema
- Interviste con gli esperti del dominio
- Conoscenza personale del dominio
- Sistemi simili già esistenti

Documentazione sui casi d'uso

- I casi d'uso sono documentati da
  - Una breve descrizione
    - Lo scopo degli use case in poche linee
  - Flusso dettagliato degli eventi
    - Descrizione dei flussi primari ed alternativi degli eventi che seguono lo start-up dello use case
  - La documentazione dovrebbe essere come un dialogo tra l'attore e lo use case
- Tutti i documenti sono scritti in modo comprensibile per il cliente

Flusso degli eventi

– Ogni caso d’uso
  - Ha una sequenza di transizioni normale o di base
  - Puo’ avere varie sequenze alternative
  - Ha varie sequenze di transazioni eccezionali per la gestione di situazioni erronee



Flusso degli eventi

- Descrive solo gli eventi relativi al caso d'uso, e non quel che avviene in altri casi d'uso
- Evita l'uso di termini vaghi come "per esempio", ″ecc." o ″informazione"
- Il flusso degli eventi dovrebbe descrivere
  - Come e quando il caso d'uso inizia e finisce
  - Quando il caso d'uso interagisce con gli attori
  - Quali informazioni sono scambiate tra un attore e il caso d'uso
    - Si tralasciano i dettagli dell'interfaccia utente
  - Il flusso di base degli eventi
  - Ogni flusso alternativo degli eventi

Casi d’uso e scenari

Scenario base: è di solito quello che prevede il *successo* del caso d’uso, ed uno svolgimento lineare

Scenari alternativi: possono essere di successo o fallimento, con complicazioni varie

- non è necessario (e sarebbe molto costoso) analizzare in dettaglio tutti i possibili scenari di un caso d’uso
- è invece necessario individuare le singole possibili varianti che possono portare al fallimento del caso d’uso, o che comportano trattamenti particolari

Come scrivere un caso d'uso

- Assegnare un nome al caso d'uso
- Descrivere gli attori
- Descrivere la condizione iniziale
- Flusso degli eventi
- Descrivere la condizione d'uscita
- Eccezioni
- Descrivere i requisiti di qualità (non funzionali)

Esempio: apertura conto corrente

1 il cliente si presenta in banca per aprire un nuovo c/c
2 l'addetto riceve il cliente e fornisce spiegazioni
3 se il cliente accetta fornisce i propri dati
4 l'addetto verifica se il cliente è censito in anagrafica
5 l'addetto crea il nuovo conto corrente
6 l'addetto segnala il numero di conto al cliente

Varianti:

3 (a) se il cliente non accetta il caso d'uso termina
3 (b) se il conto va intestato a più persone vanno forniti i dati di tutte
4 (a) se il cliente (o uno dei diversi intestatari) non è censito l'addetto provvede a registrarlo, richiede al cliente la firma dello specimen e ne effettua la memorizzazione via scanner

Esempio: apertura conto corrente - ulteriore dettaglio

(5) l'addetto crea il nuovo conto corrente

1 l'addetto richiede al sistema la transazione di inserimento nuovo conto

2 il sistema richiede i codici degli intestatari

3 l'addetto fornisce i codici al sistema

4 il sistema fornisce le anagrafiche corrispondenti, e richiede le condizioni da applicare al conto

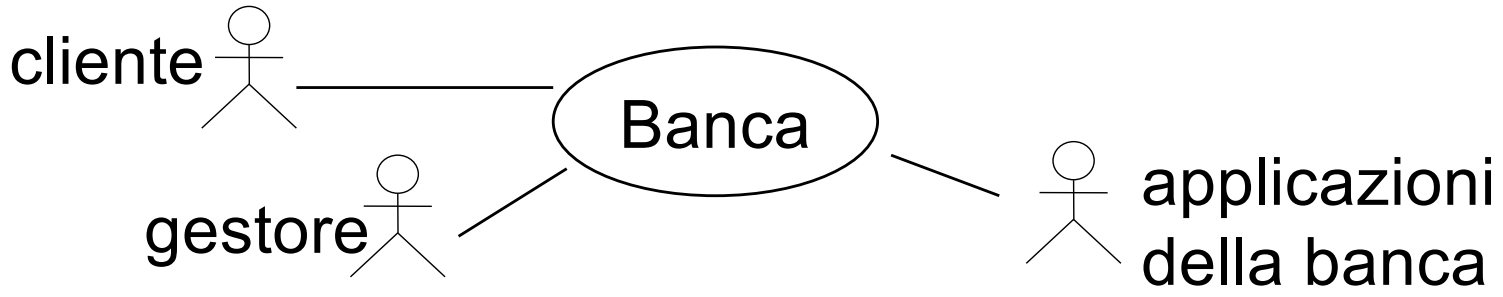
5 l'addetto specifica le condizioni e chiede l'inserimento

6 il sistema stampa il contratto con il numero assegnato al conto

Varianti:

3 (a) se il sistema non riconosce il cliente, o se fornisce un'anagrafica imprevista, l'addetto può effettuare correzioni o terminare l'inserimento

Dalle funzionalità interne ai casi d'uso



casi d'uso

- cliente:
  - disposizioni (bonifici, acquisto titoli, ecc.)
  - informativa
  - stipula contratto
- gestore:
  - verifica anomalie

funzionalità interne sistema

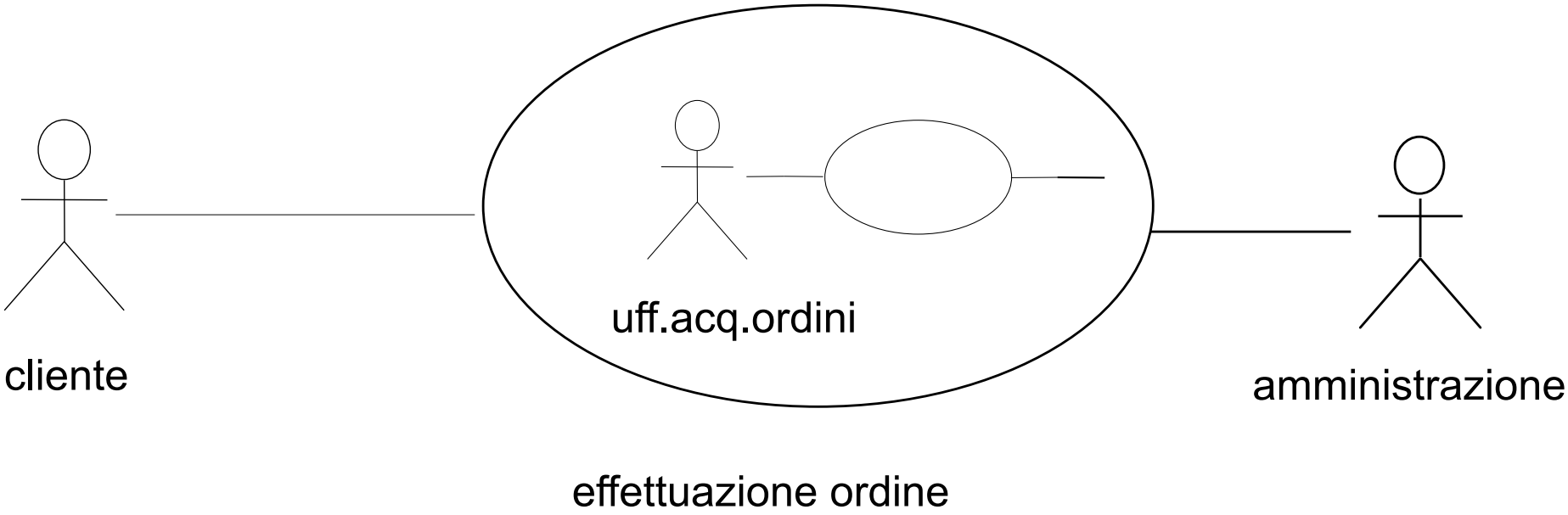
- front-end specializzati
- front-end comune
- gestione pre-applicativa
- gestione contratti
- monitoraggio sistema

Attori “business” e “di servizio”

Nel definire gli attori si possono adottare due diversi punti di vista che corrispondono a diversi livelli di astrazione:

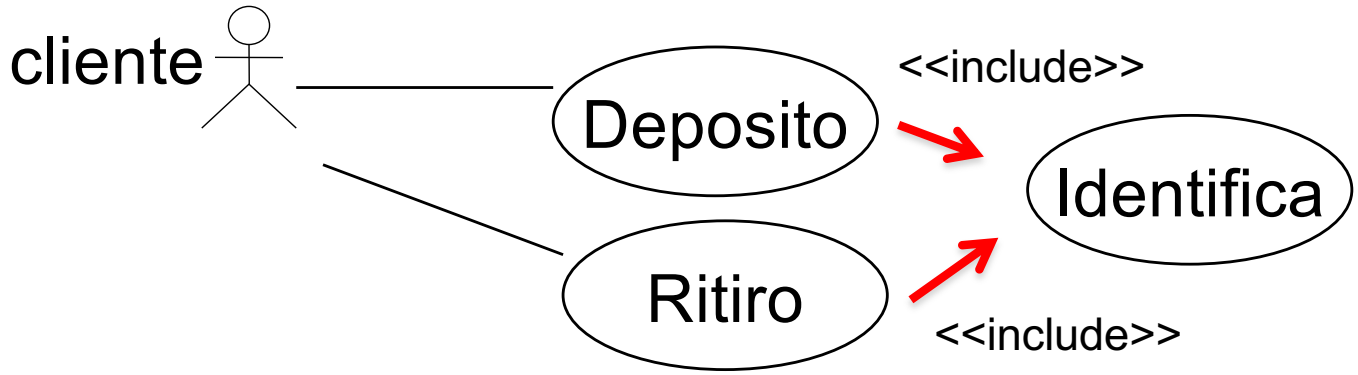
- uno indipendente da particolari soluzioni organizzative e tecnologiche (modello dell’ambito "business workflow")
- uno legato ad una particolare soluzione organizzativa e tecnologica (modello dei servizi del sistema informatico)

Modellazione dell'ambito di business



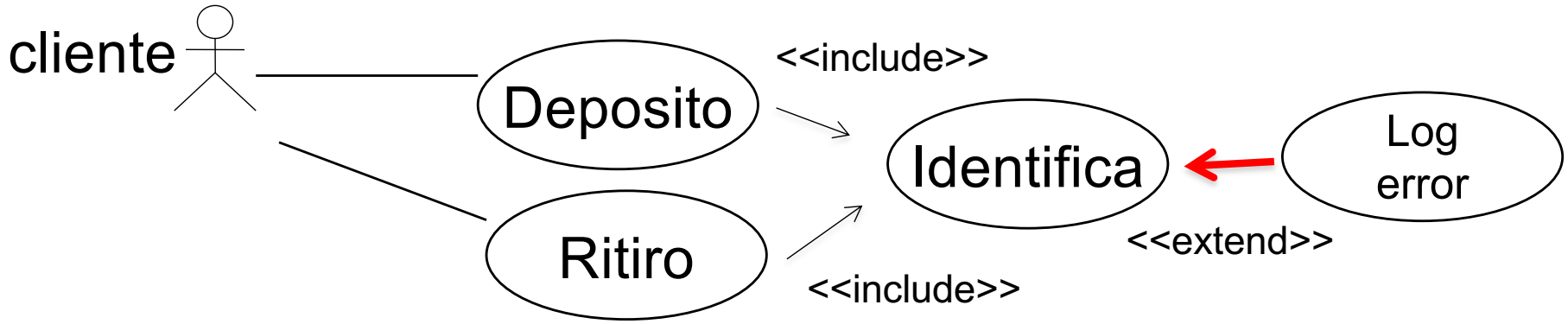
Dal punto di vista del cliente, l'ufficio acquisizione ordini fa parte del sistema (come intermediario)

Includere un caso d'uso



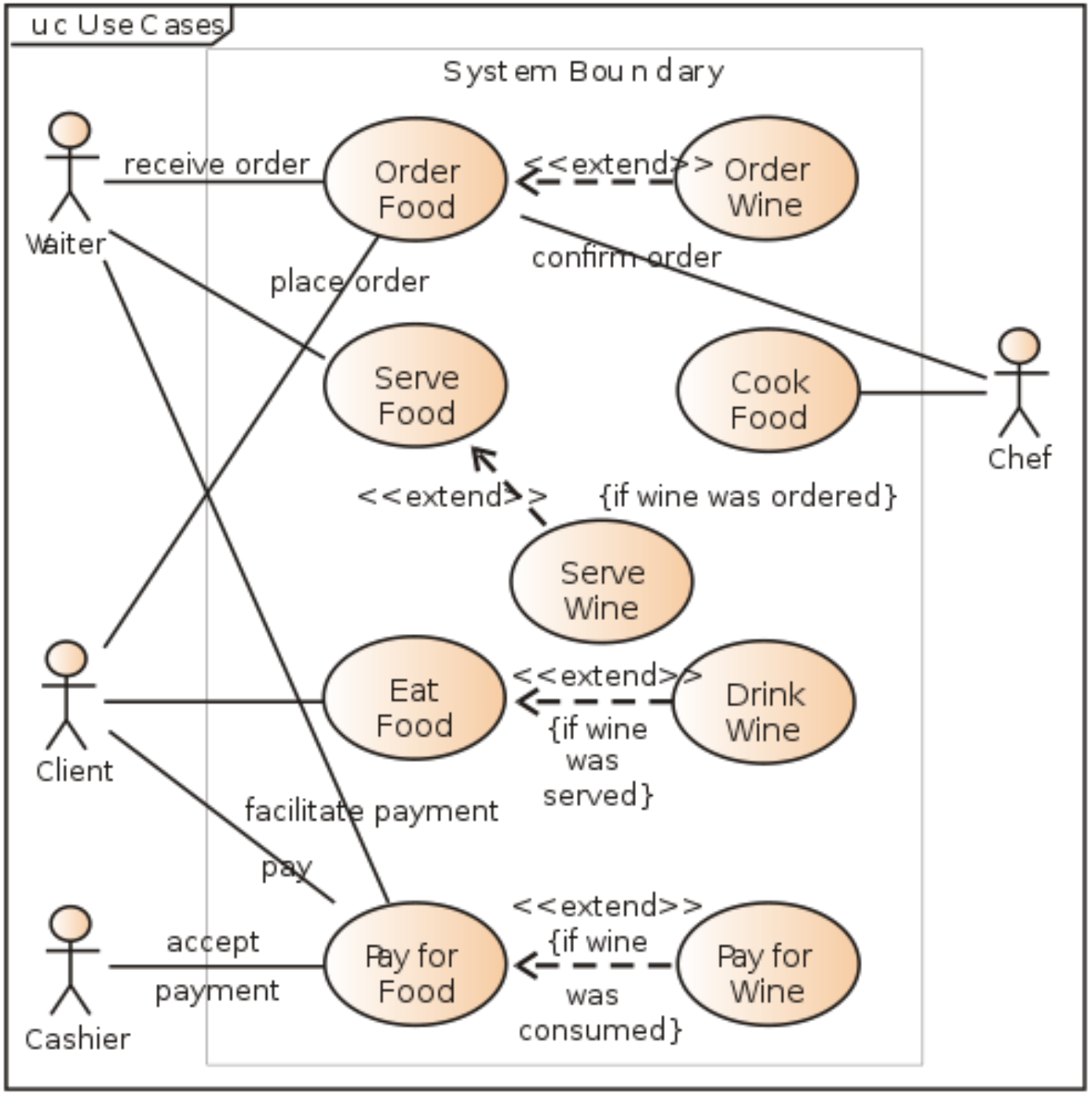
- Un caso d'uso "include" il comportamento di un altro caso d'uso
- (simile a "chiamata di procedura")
- Notare il verso della freccia: "Deposito" e "Ritiro" includono "Identifica"

Estendere un caso d'uso



- Un caso d'uso "estende" il comportamento di un altro caso d'uso
- (simile a "funzione opzionale")
- Notare il verso della freccia: "Log error" estende "Identifica"

Esempio


uc Use Cases
System Boundary
Waiter
receive order
Order Food
<<extend>>
Order Wine
confirm order
place order
Serve Food
Cook Food
Chef
<<extend>>
{if wine was ordered}
Serve Wine
Client
Eat Food
<<extend>>
Drink Wine
{if wine was served}
facilitate payment
pay
Cashier
accept payment
Pay for Food
<<extend>>
{if wine was consumed}
Pay for Wine

Analisi dei requisiti

- 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.

Regole per scrivere il documento dei requisiti

- Definire il glossario (nomi e verbi)
- Unificare i termini sinonimi
- Specificare ciò che è troppo generico
- Introdurre eventuali elementi mancanti
- Specificare lo scenario delle operazioni
- Usare i verbi al futuro

Glossario

termine	descrizione	dati
Libro o volume	Elemento della biblioteca	ISBN, titolo, autori, casa editrice, anno edizione, voce di classificazione
Catalogo	Lista dei libri catalogati mediante il sistema di classificazione	Libro, classificazione, collocazione
iscritto	Utente della biblioteca	numero tessera, cognome, nome, dati nascita, indirizzo
prestito	Cessione di un libro ad un iscritto con l'impegno di restituirlo entro un certo giorno	id. libro, id. iscritto, data cessione, data restituzione

Lista delle operazioni (scenari della qualità)

id	operazione	frequenza	agente
1	Inserimento nuovo libro	50/settimana	bibliotecario
2	Iscrizione nuovo utente	5/settimana	bibliotecario
3	Prestito di un libro	200/giorno	bibliotecario
4	Restituzione di un libro	180/giorno	bibliotecario
5	Controllo prestiti scaduti	5000/giorno	bibliotecario
6	Consultazione catalogo	1000/giorno	tutti

Il documento dei requisiti

- L’applicazione gestirà i prestiti dei volumi agli iscritti.
- L’applicazione dovrà usare il sistema di classificazione per descrivere il contenuto di ciascun volume con i relativi metadati bibliografici (autore, titolo, editore, anno, ISBN) e associare un contatore di copia.
- L’applicazione permetterà l’accesso via web a) ai bibliotecari per aggiornare la base di dati, b) agli utenti per consultare i metadati dei volumi.
- …

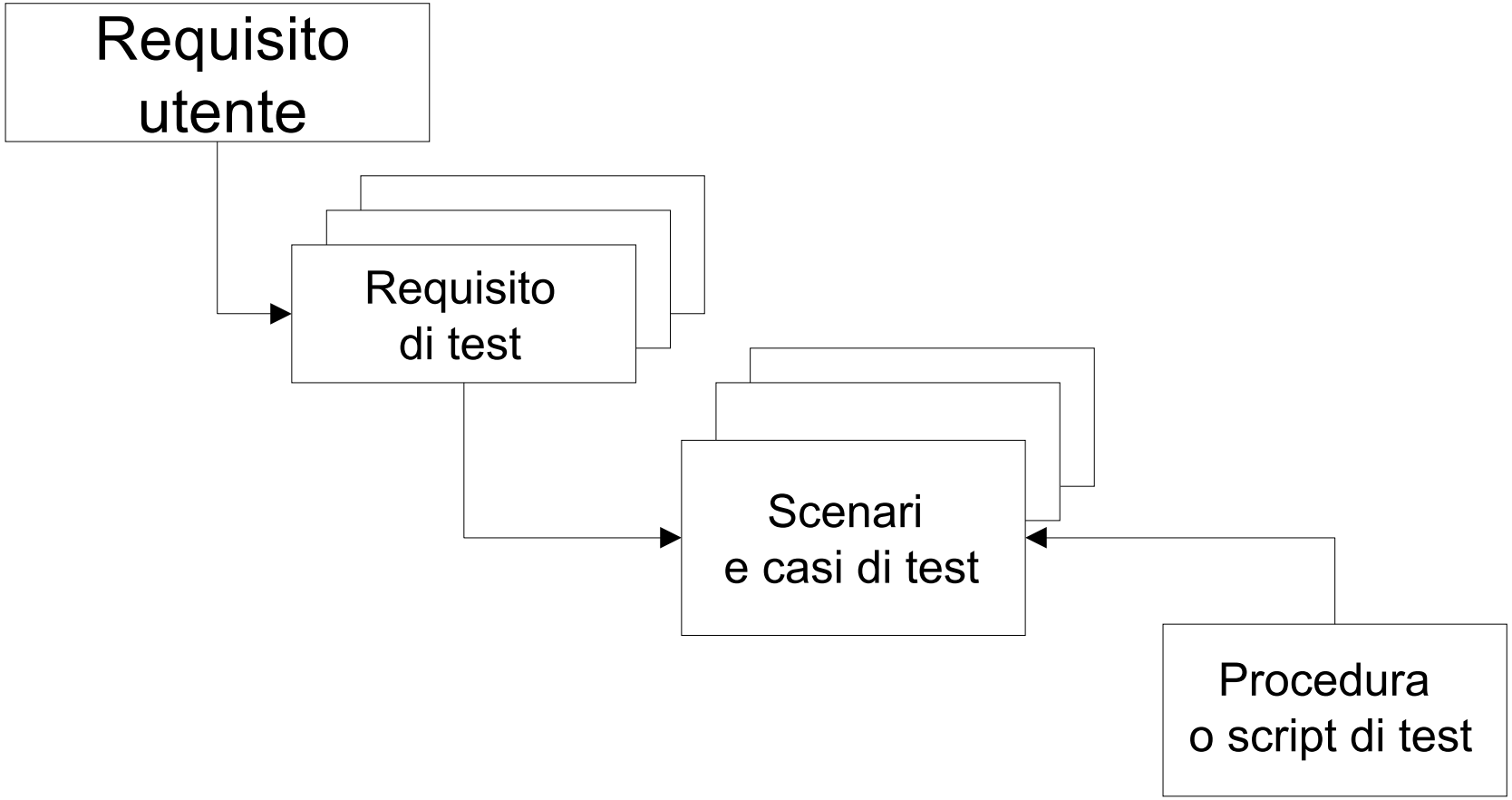
Requisiti e test

- Ogni requisito deve avere un risultato osservabile
- Ogni requisito dovrà trovare “soddisfazione” nel codice
- Il codice dovrà essere controllato (*testato*) per “verificare” che soddisfi il requisito per cui è stato scritto

Esempio

- Requisito: l'applicazione gestirà i prestiti dei volumi agli iscritti
- Possibile risultato osservabile: quando qualcuno chiede un libro in prestito bisogna registrare la richiesta e se il volume è disponibile dare il libro e registrare il prestito con la scadenza
- Possibile test:
  `richiesta_prestito("utente_Paolo","Promessi sposi")`

Requisiti e test



Esempio: bancomat

Requisiti:

1. “Il bancomat permetterà di prelevare denaro”
2. “Si potranno prelevare da 20€ a 300€”
3. “I prelievi saranno in multipli di 20€”

Esercizio:

- Solo tre requisiti.
- Cosa testiamo?
- Ci sono dei requisiti impliciti che occorre esplicitare?

Testare i prelievi

Requisiti sui test:

"Si controllerà che esista l'opzione di prelievo"

"Si controllerà di poter fare prelievi di multipli di 20€, tra 20€ e 300€"

"Si controllerà che sia proibito prelevare <20€"

" Si controllerà che sia proibito prelevare >300€"

" Si controllerà che sia proibito prelevare multipli di 20€ > 300€ "

" Si controllerà che sia proibito prelevare non-multipli di 20€"

" Si controllerà che siano proibite combinazioni strane (es.: tanti zeri, tanti 9, 19.999999)"

"Si controllerà che il denaro prelevato sia pari a quello richiesto"

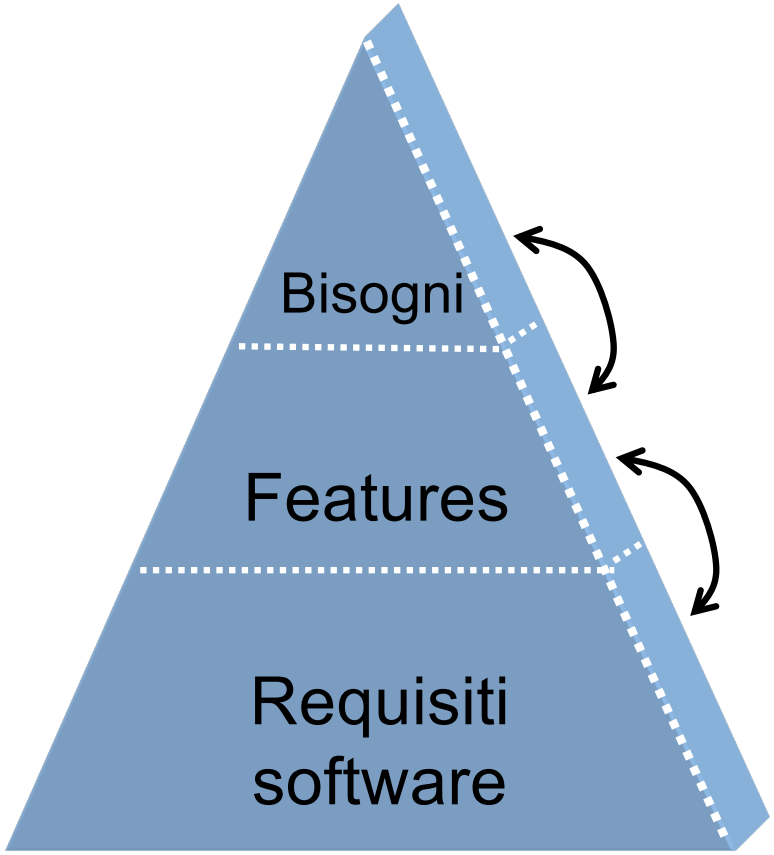
" Si controllerà che il prelievo sia coperto dal saldo esistente nel conto"

- *Questi sono requisiti sui test e NON test perché non descriviamo i dati da usare (es: 21, 40, 60, 9999)*
- *I dati verranno decisi quando eseguiremo i casi di test*

La piramide dei requisiti

In cima sono i *bisogni* del cliente.
Esempio: “I dati dei clienti dovranno essere conservati per 10 anni".

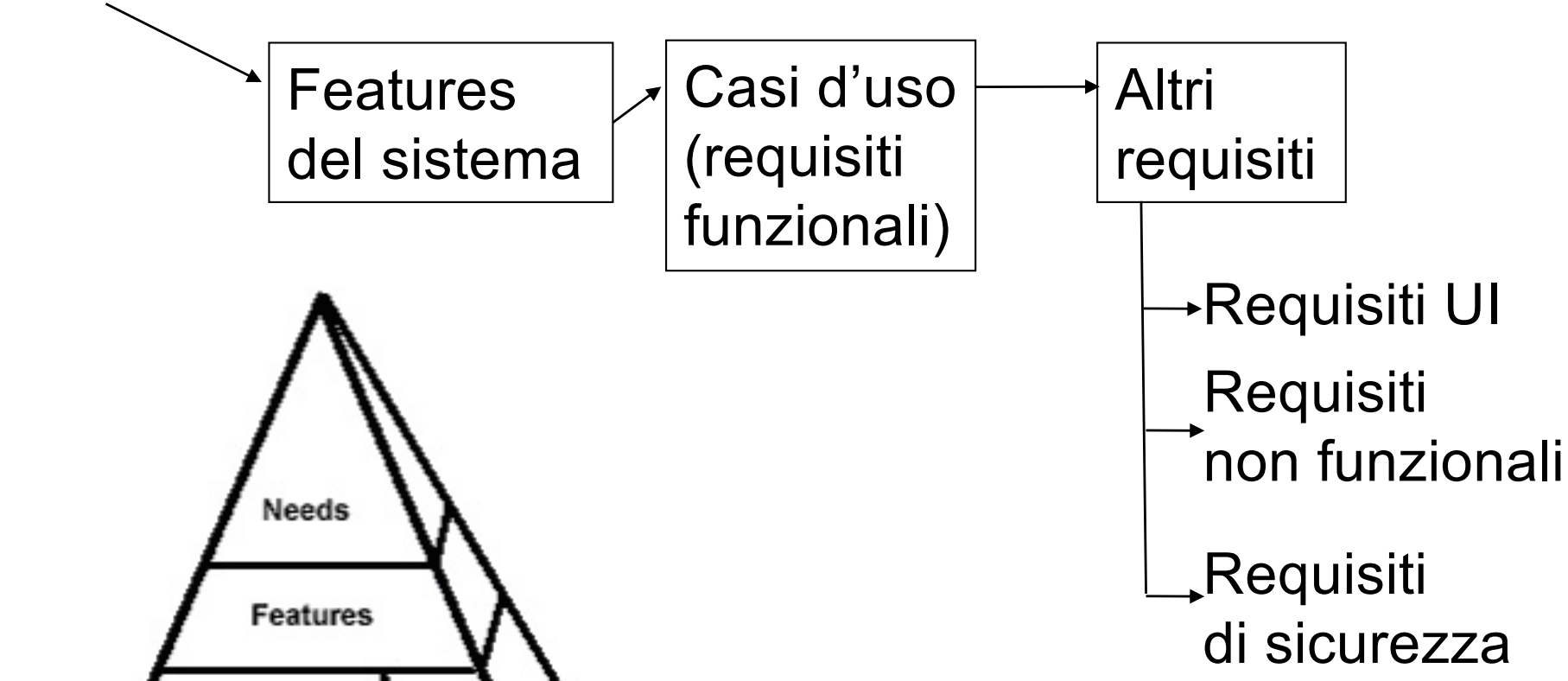
Al bisogni danno risposta le “features”: ogni feature corrisponde ad un bisogno ed ha proprietà (requisiti funzionali o meno) che il sistema finito dovrà possedere e che verranno controllate mediante test



www.ibm.com/developerworks/rational/library/04/r-3217

Dai bisogni ai requisiti

Bisogni dell'utente



Feature

Una *feature* è un servizio che soddisfa uno o più bisogni del cliente.

Per esempio: "*Il sistema offre un database relazionale per gestire i dati persistenti*"

La nozione di *feature* è utile specie quando si deve trattare con sistemi sw fatti di componenti riusabili, cioè riconfigurabili o modificabili

Software Features	Pro	Suite	Plot	Pipes	SetUps	Dev
PIdot	x	x	x	x	x	x
Envelope	x	x	x	x	x	x
Lines 5/1 5/2 5/3 5/9 6/1 6/5 1-1h/l	x		x		x	x
Lines 5/2ex 5/9ex 6/6 6/7	x		x		x	x
Flow	x		x		x	x
Dot Refresh	x	x	x		x	x
Main Channel Line	x		x		x	x
Trading Types	x		x		x	x
Support/Resistance Nearby & Furtherout	x		x		x	x
Time Frames Any 3 Intraday + 5 Daily & Higher	x	x	x	x	x	x
Higher Time Frame Overlays on Lower Time Frame	x	x	x	x	x	
Walk Forward Testing	x	x	x	x	x	
Full Control of TradeStation Paint Layers			x			
Full Accuracy Daily on Intraday Plotting	x	x	x	x	x	x
Refresh Bands Including Higher Time Period Overlays		x	x		x	
Blues Including Higher Time Period Overlays		x	x		x	
Mother Goose Including Higher Time Period Overlays		x	x		x	
59zones Including Higher Time Period Overlays		x	x		x	
Pipes Including Higher Time Period Overlays			x	x	x	
Quarterly, Yearly and Multi-Year Bar Charts & Overlays		x				
Bar Pct Complete Utility		x				
Alerts - Price Nears Any Combination of DG lines					x	
Alerts - Radar Screen Scanning					x	
Set-Ups - Pre Determined Price & Structure					x	
Complete DG2 Data Base Access			x		x	x
New Concepts Such As:			x		x	x
Rvals - Values for Structure Relationships			x		x	x
WIE - Where is Envelope			x		x	x
AD - Accumulation & Distribution Indication			x		x	x
1/4/2011						
EL Function Full Access to DG2 Database						x

Feature

IEEE 829 (*Standard for Software and System Test Documentation*):

Feature: caratteristica distintiva di un componente software

Esempio: in MS Word, sono feature i meccanismi di editing, di gestione degli indici, di impaginazione, ecc.

Esempi di *feature*

- Feature: *Il prodotto include un ambiente speciale per editare formule matematiche*
- Feature: *Il prodotto può impaginare documenti o con layout predefiniti o definiti da utente*
- Feature: *Il prodotto può importare un documento in formato RTF*

Requisito (requirement)

- Una frase che definisce il risultato di una funzione o processo, utilizzabile da qualcuno
- Esempio:

Req. 3.4.B The database shall support the generation and control of configuration objects; that is, objects which are themselves groupings of other objects in the database. The configuration control facilities shall allow access to the objects in a version group by the use of an incomplete name

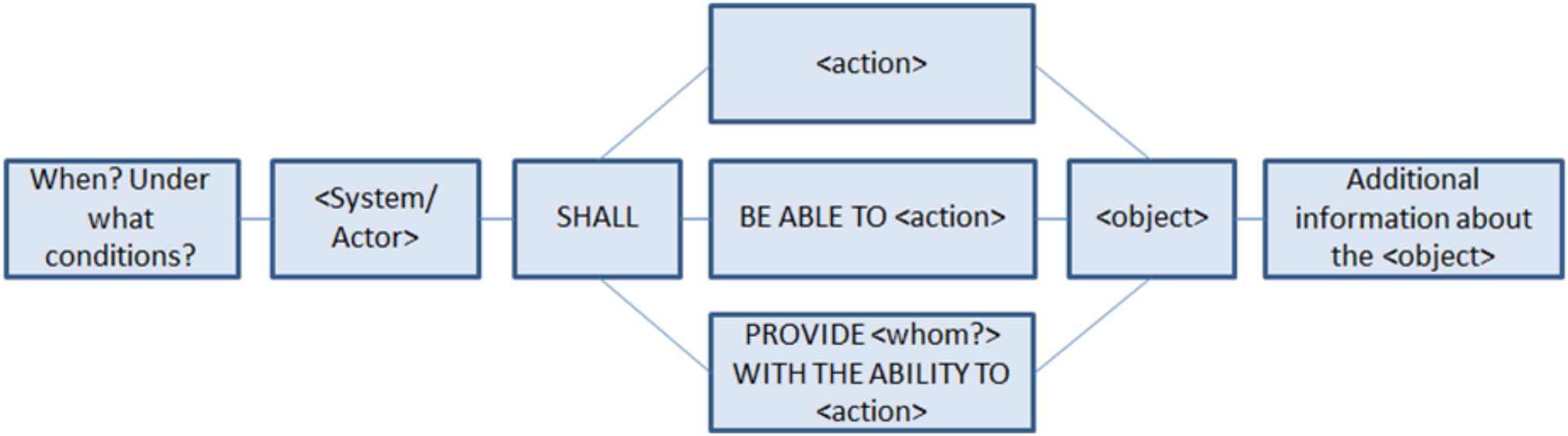
Una lista di requisiti

Requirements Document

REQ001	System shall have fields for firstname, middle initial and last name.
REQ002	System shall display a name if there is one in the stored profile.
REQ003	System shall require name is completed.
REQ004	System shall have a field for position or title.
REQ005	System shall require title is completed.
REQ006	System shall display a position or title if there is one in the stored profile.
REQ007	System shall have a field for email address.
REQ008	System shall have a field for alternate email address.
REQ009	System shall display an email address if there is one in the stored profile.
REQ010	System shall display an alternate email address if there is one in the stored profile.
REQ011	System shall require email address is completed.
REQ012	System shall require alternated email address is completed.
REQ013	System shall have a field for daytime phone number.
REQ014	System shall display a phone number if there is one in the stored profile.
REQ015	System shall require phone number is completed.
REQ016	System shall validate all characters in the phone number field are digits when user exits the field.
REQ017	System shall display an error message if not all characters in the phone number field were digits.
REQ018	System shall have a field for a fax number.
REQ019	System shall require fax is completed.
REQ020	System shall display a fax number if there is one in the stored profile.
REQ021	System shall validate all characters in the fax number field are digits when user exits the field.
REQ022	System shall display an error message if not all characters in the fax number field were digits.
REQ023	System shall have two fields for a street address.
REQ024	System shall require the first street address field is completed.
REQ025	System shall display an address if there is one in the stored profile.
REQ026	System shall have a field for city.
REQ027	System shall require the city field is completed.
REQ028	System shall display a city if there is one in the stored profile.
REQ029	System shall have a field for state.
REQ030	System shall display a state if there is one in the stored profile.
REQ031	System shall require the state field is completed.
REQ032	System shall have a field for zip code.
REQ033	System shall display a zip code if there is one in the stored profile.
REQ034	System shall require the zip code field is completed.

I requisiti sono frasi strutturate, cioè seguono uno schema prefissato

La International Requirements Engineering Board (IREB) ha definito uno schema – detto *boilerplate* – per dare forma standard ai requisiti:



Esempio: Quando la biblioteca è aperta un iscritto potrà consultare il catalogo in cui potrà cercare i volumi disponibili

Che cos'è un requisito?

- Un requisito può essere una **descrizione** astratta e generica di un servizio o vincolo di sistema, oppure una **specifica** concreta e dettagliata di una funzionalità del sistema
- I requisiti hanno infatti una doppia funzione:
  - Base per un'*offerta di contratto*: devono essere aperti a varie interpretazioni
  - Base per un *contratto*: devono essere dettagliati e non soggetti ad interpretazioni arbitrarie

A Software Requirement is … (IEEE 610)

1. A condition or capability needed by a user to solve a problem or achieve an objective
2. A condition or capability that must be met or possessed by a system or component to satisfy a contract, standard, specification, or other formally imposed documents
3. A documented representation of a condition or capability as in (1) or (2)

Cosa sono i requisiti?

I requisiti sono **proprietà** di un sistema o prodotto **desiderate** dai suoi committenti e **verificabili**

- Quali proprietà?
  - funzionali oppure non-funzionali
- Che succede se alcuni requisiti confliggono?
  - Priorità / preferenze
- Chi sono i committenti e come si registrano i relativi requisiti?
  - Stakeholders che usano viste specializzate
- Come si verifica che un requisito sia soddisfatto?
  - Testing (sviluppatori)
  - Validazione (committenti)

Esempi di requisiti

- Requisito: *l'editor permetterà di gestire testi matematici*
- Requisito: *l'editor permetterà di impaginare automaticamente la lista delle figure e l'indice analitico*
- Requisito: *L'editor potrà interoperare con un programma di foglio elettronico importando le tabelle*

Differenza tra feature e requisito

- Un requisito descrive una capacità che deve possedere il sistema; è scritto per testare l'implementazione in relazione agli scenari di uso del sistema stesso
- Una feature è un insieme di requisiti logicamente correlati; di solito descrive una funzionalità a grana grossa di un prodotto
- Esempio:
  - *Feature*: carrello in un sito di e-commerce
  - *Requisiti*:
    - L'utente potrà aggiungere elementi al carrello
    - L'utente potrà rimuovere elementi dal carrello
    - L'utente potrà iniziare il pagamento dal carrello
    - …

Il documento dei requisiti

SRS:
software requirements specification document

Table of Contents

I requisiti sono “bifronti”

Se un’organizzazione deve assegnare un **contratto** per lo sviluppo di un sistema software, scrive un documento (“capitolato”) per definire i suoi scopi in modo sufficientemente astratto da non predisporre una particolare soluzione e poter esporre un bando

I requisiti devono essere scritti in modo che più possibili contraenti possano rispondere al bando, offrendo diversi modi di realizzare la soluzione coerentemente col capitolato

Dopo che il contratto è stato assegnato, il contraente vincitore deve scrivere per il cliente un “documento tecnico” più dettagliato rispetto al capitolato, che possa essere allegato al contratto

Entrambi i documenti (capitolato e documento tecnico) possono essere definiti “*documento dei requisiti*” del sistema da sviluppare

Specifica e definizione dei requisiti

- **Definizione dei requisiti**
  - Descrizione in linguaggio naturale dei bisogni e dei vincoli operativi di un sistema
  - Si scrive per i clienti
- **Specifica dei requisiti**
  - Documento strutturato che dettaglia i servizi attesi dal sistema
  - Scritto come contratto tra cliente e contraente
  - Si scrive per gli sviluppatori

Domande utili per scrivere i requisiti

- Cosa deve fare il software?
- Che interfacce ha con i suoi utenti, con l'hw, con altri prodotti sw?
- Che prestazioni deve esibire il software?
- Quali attributi (es. interfaccia accessibile) dovrà avere?
- Quali vincoli dovrà soddisfare?

Cosa non è un requisito?

- L'architettura del sistema
- Vincoli tecnologici (es. linguaggio di implementazione)
- Il processo di sviluppo
- L'ambiente di sviluppo
- Il sistema operativo di riferimento
- Aspetti di riusabilità e portabilità

- **Nota**: le descrizioni del dominio del mondo reale fanno parte dei requisiti (anche se il cliente non le richiede)

Tipi di requisito

- **Requisiti funzionali**: Descrivono le interazioni tra il sistema ed il suo ambiente, indipendentemente dall'implementazione
  - La piattaforma e-learning tiene traccia delle attività dello studente

- **Requisiti non funzionali**: Proprietà del sistema misurabili dall'utente e non direttamente correlate al suo comportamento funzionale
  - La piattaforma deve gestire almeno 100 utenti contemporaneamente
  - Se un utente fa una domanda per email deve avere risposta entro 48h
  - La piattaforma deve essere disponibile agli studenti 24/7

- **Vincoli** ("Pseudo requisiti"): Imposti dal cliente o dall'ambiente operativo in cui funzionerà il sistema
  - Il linguaggio di programmazione dev'essere Java
  - La piattaforma deve interfacciarsi con documenti scritti con Word su Windows XP

Requisiti funzionali

- Un requisito funzionale definisce una funzione di un sistema, inclusi il suo ingresso e la sua uscita
- I requisiti funzionali determinano lo sviluppo di codice
- Il testing del codice rispetto ai requisiti funzionali costituisce la **verifica**

Esempi di requisiti funzionali

- *L'utente dovrà essere in grado di cercare o in tutti i database o in un loro sottoinsieme*
- *Il sistema fornirà i visualizzatori appropriati in modo che l'utente possa leggere i documenti archiviati*
- *Ad ogni fattura verrà associato un identificatore unico (ORDER_ID) che l'utente potrà copiare nell'area di memoria permanente del suo conto*

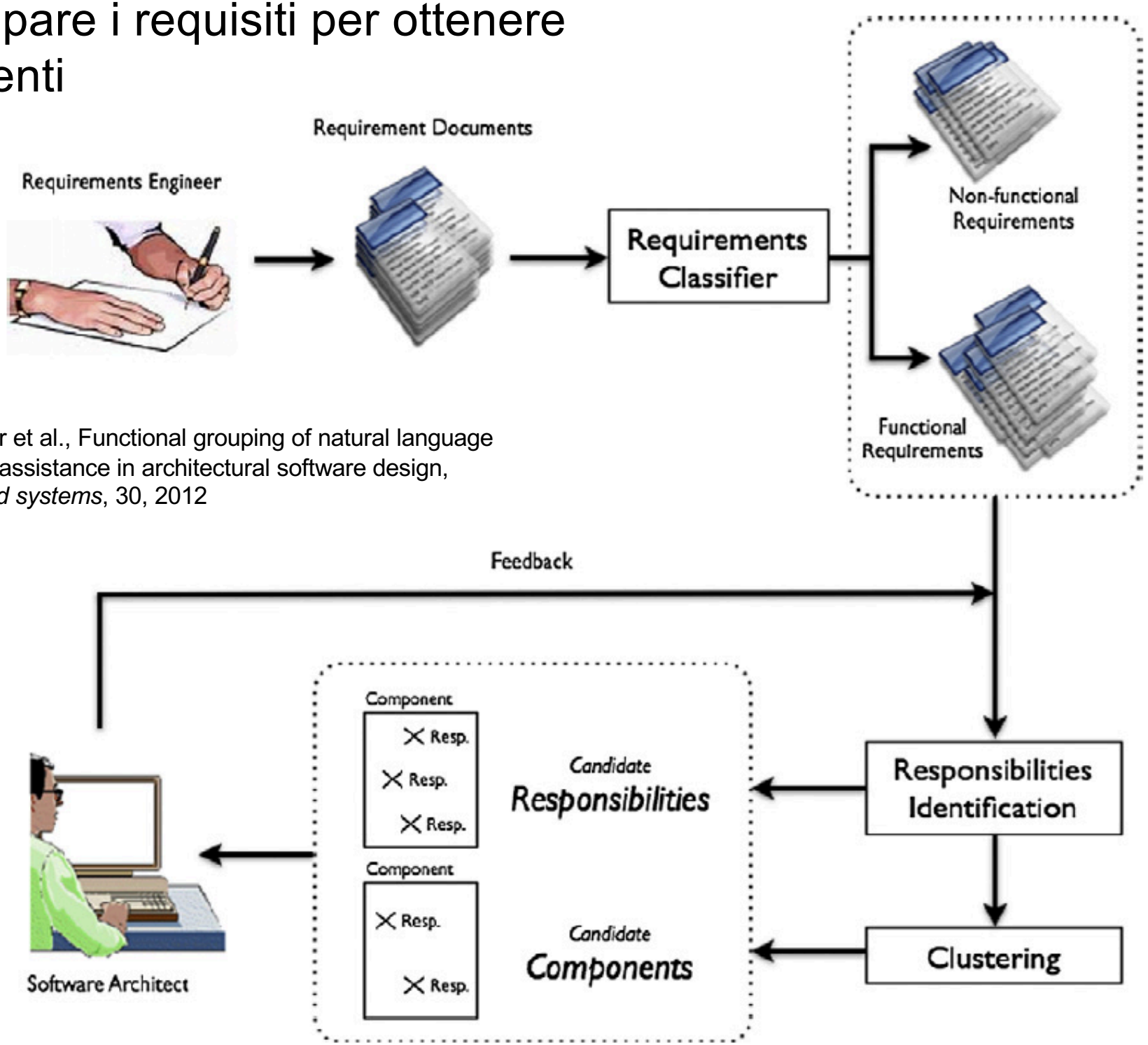
Requisiti non funzionali

- I requisiti non funzionali sono di solito più critici di quelli funzionali: se non vengono soddisfatti il sistema non verrà utilizzato
- I requisiti non funzionali definiscono proprietà del sistema o vincoli sul suo sviluppo
  - Esempi di proprietà prestazionali: tempo di esecuzione, tempo di risposta, dimensioni della memoria necessaria, throughput.
  - Esempi di proprietà di "*dependability*": affidabilità, robustezza, sicurezza, integrità, manutenibilità
  - Esempi di vincoli : funzionalità di dispositivi I/O, tipo di processore, ecc.
- Si possono anche specificare vincoli di processo, cioè sul metodo di sviluppo, sul linguaggio di programmazione, sugli strumenti da usare

Esempi di requisiti non funzionali

- Requisito di prodotto
  - Quando l'utente ha scelto il tipo di biglietto la transazione dev'essere completata entro due secondi
  - Per ogni messaggio dovrà essere possibile usare l'insieme di caratteri standard ASCII
- Requisito organizzativo
  - Il processo di sviluppo ed i documenti consegnati saranno conformi al processo ed alle strutture di documento descritte nello standard ISO-XYWZ-2007
- Requisito esterno
  - Il sistema non permetterà ai suoi operatori di conoscere alcuna informazione personale sui clienti, eccetto il nome ed il numero di riferimento interno

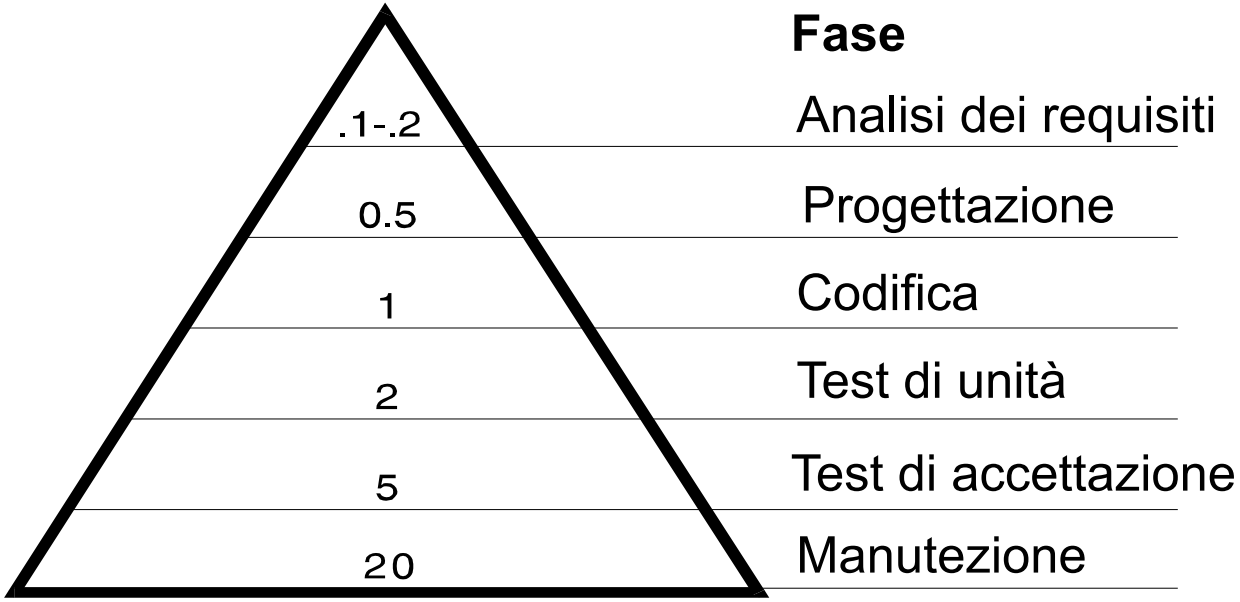
Raggruppare i requisiti per ottenere componenti



fonte: Casamayor et al., Functional grouping of natural language requirements for assistance in architectural software design, *Knowledge-based systems*, 30, 2012

Sbagliare i requisiti è costoso

Trovare gli errori durante la fase dei requisiti può far risparmiare fino a 200 volte rispetto allo stesso errore trovato durante la manutenzione



Barry Boehm- ‘76, 88

Oltre metà di tutti i difetti di un sistema sw possono essere attribuiti ad errori fatti durante la stesura dei requisiti

Attività di elicitazione dei requisiti

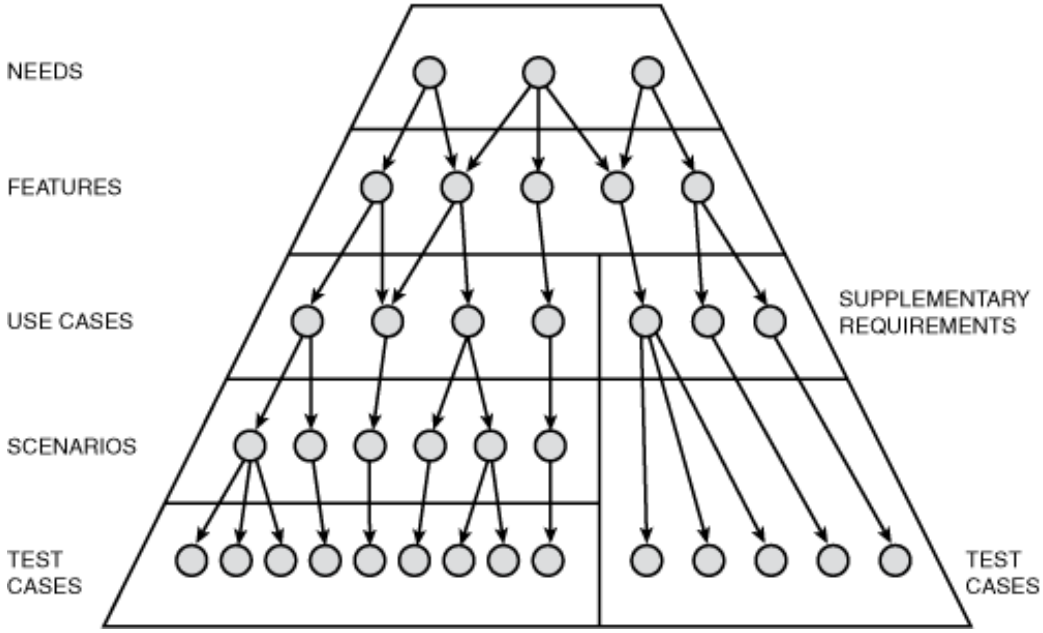
- Identificare gli attori
- Identificare i casi d’uso
- Identificare gli scenari
- Identificare le relazioni tra i casi d’uso
- Raffinare i casi d’uso
- Identificare i requisiti non funzionali
- Identificare gli oggetti partecipanti

Attività di analisi dei requisiti

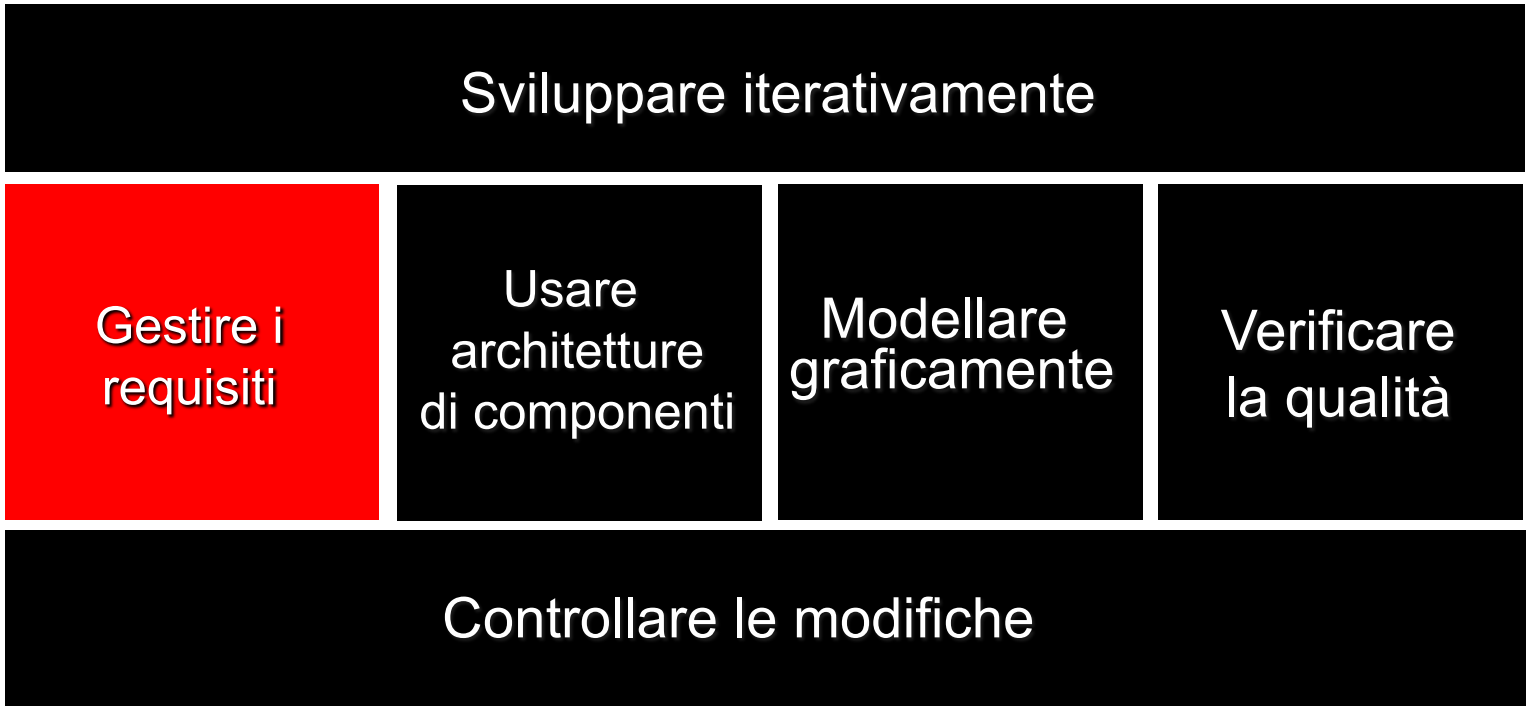
- Classificare
- Chiarire
- Normalizzare
- Ordinare
- Tracciare

Dai bisogni degli utenti al test

- Un requisito è una funzionalità che il sistema finito dovrà esibire e che verrà verificata (con test)
- Una *feature* è un servizio fornito dal sistema per soddisfare uno o più bisogni (requisiti) del cliente
- Un caso d'uso descrive una sequenza di azioni, eseguite da un sistema, per produrre un risultato che ha valore per un utente
- I casi di test vengono eseguiti su scenari di casi d'uso



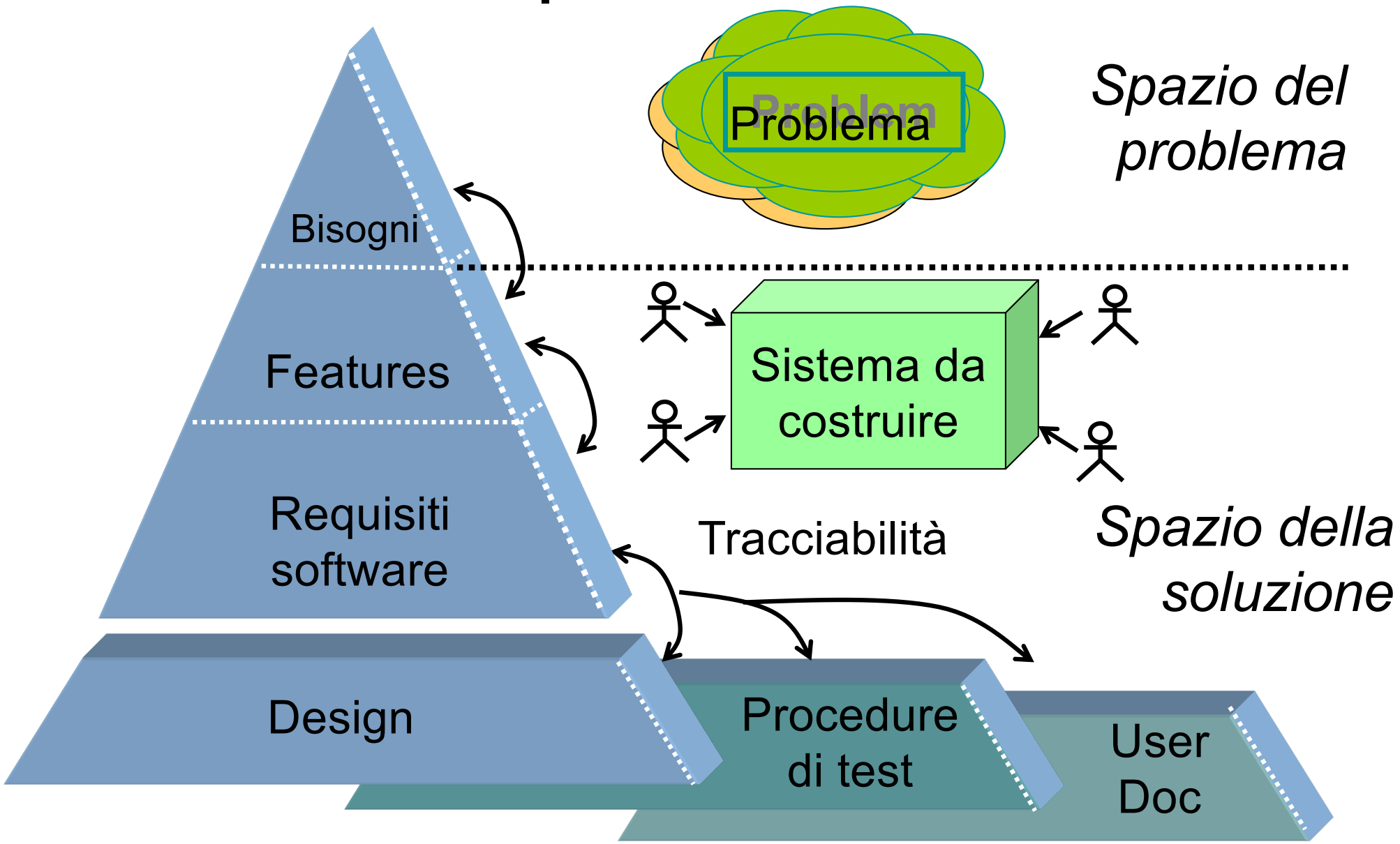
Principi guida dello sviluppo software



Gestione dei requisiti (requirements management)

- Gli strumenti di gestione dei requisiti permettono di creare un database dei requisiti
- Le operazioni sui requisiti sono: la stesura del requisito, il suo tracciamento, il controllo delle versioni, il testing
- Esistono molti strumenti, anche open source (es. rmtoo, vedi rmblog.accompa.com/2012/04/free-open-source-requirements-management-tool/)

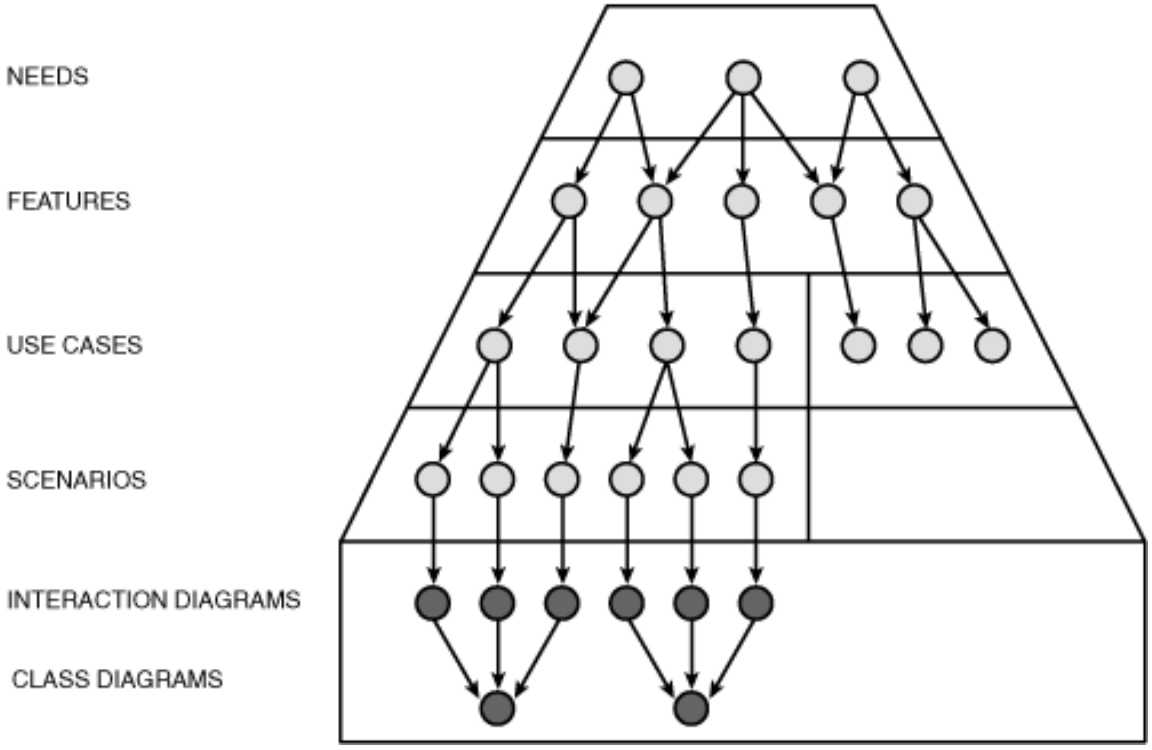
Gestire i requisiti



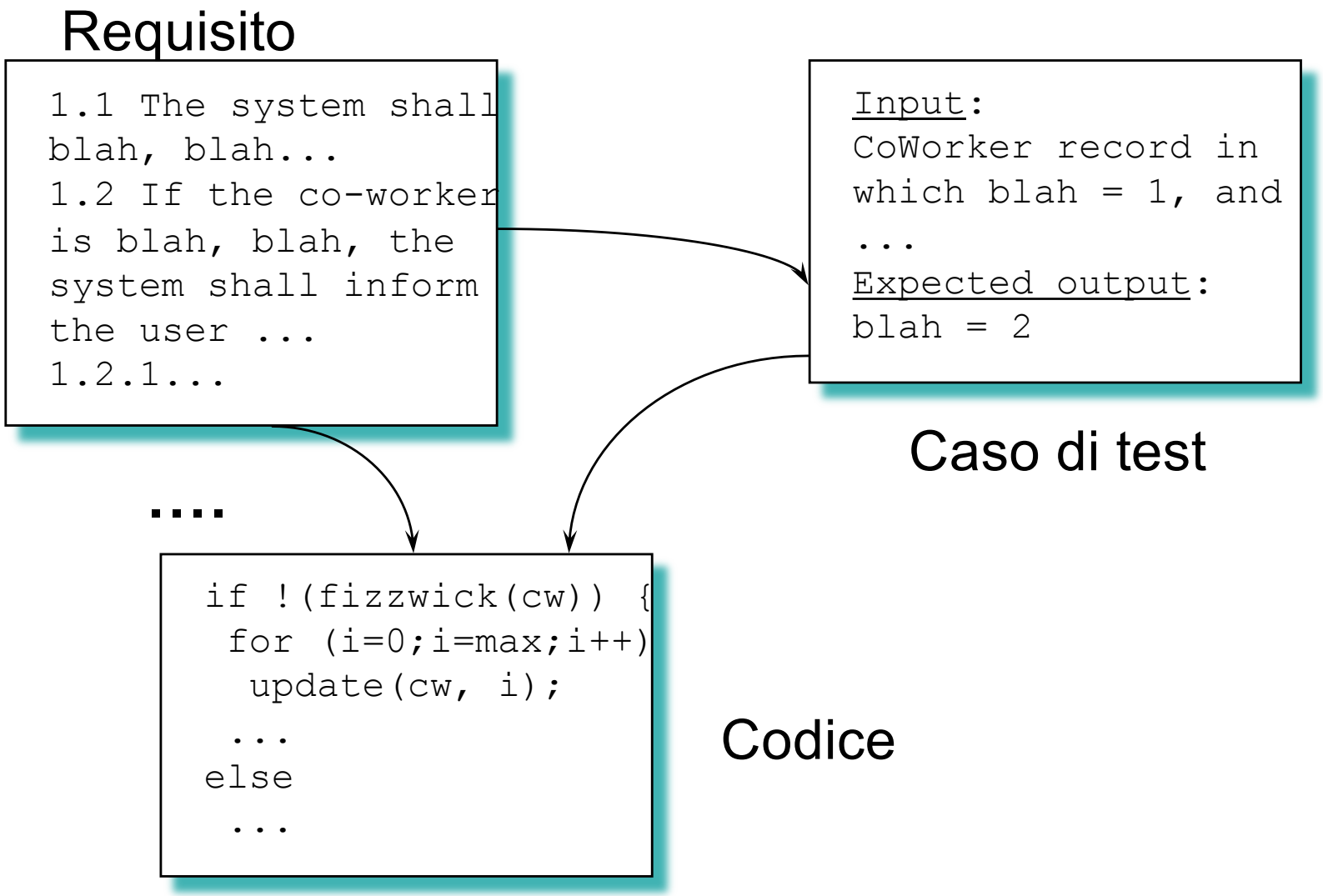
Tracciabilità dei requisiti funzionali

La tracciabilità serve a:

- Verificare che il sistema soddisfi **tutti** i requisiti del cliente
- Verificare che il sistema soddisfi **solo** i requisiti del cliente
- Gestire le modifiche dei requisiti in corso d'opera – quando cambia un requisito, e dunque il codice, dovremo rifare i test ad esso relativi



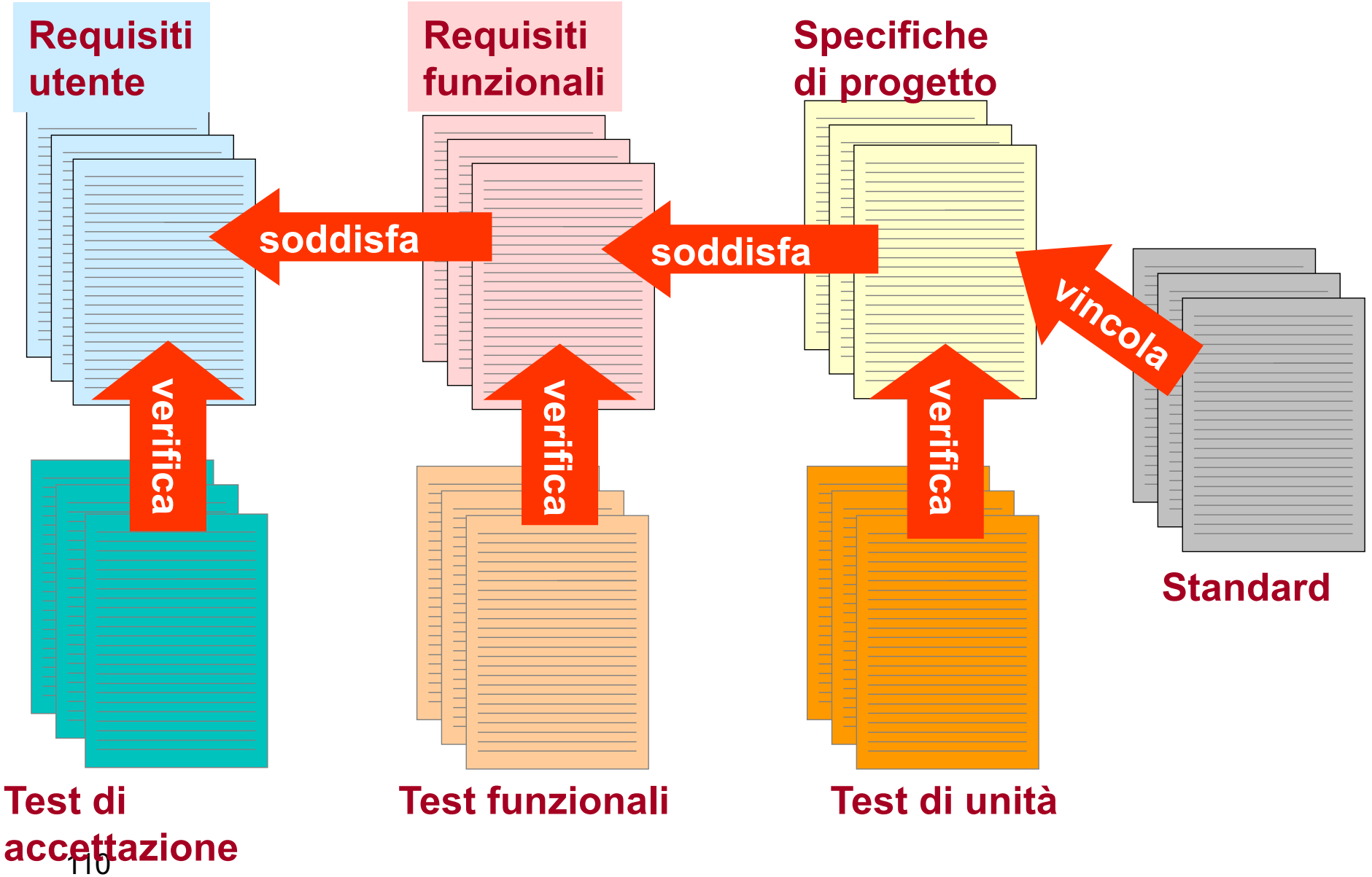
Tracciabilità



Tracciamento dei requisiti

- La tracciabilità dei requisiti è la capacità di descrivere e seguire il ciclo di vita di un requisito funzionale dall'origine al suo uso e viceversa
- La tracciabilità documenta le relazioni tra gli artefatti creati durante lo sviluppo, quali i requisiti, le specifiche, i modelli, i test ed i componenti sviluppati
- La relazione principale che riguarda i requisiti nell'ambito del processo di sviluppo è il "*soddisfacimento*": come viene soddisfatto un requisito dagli artefatti di sviluppo?
- Altra relazione è "*verifica*": come si verifica con un test che un requisito è stato soddisfatto?

La tracciabilità

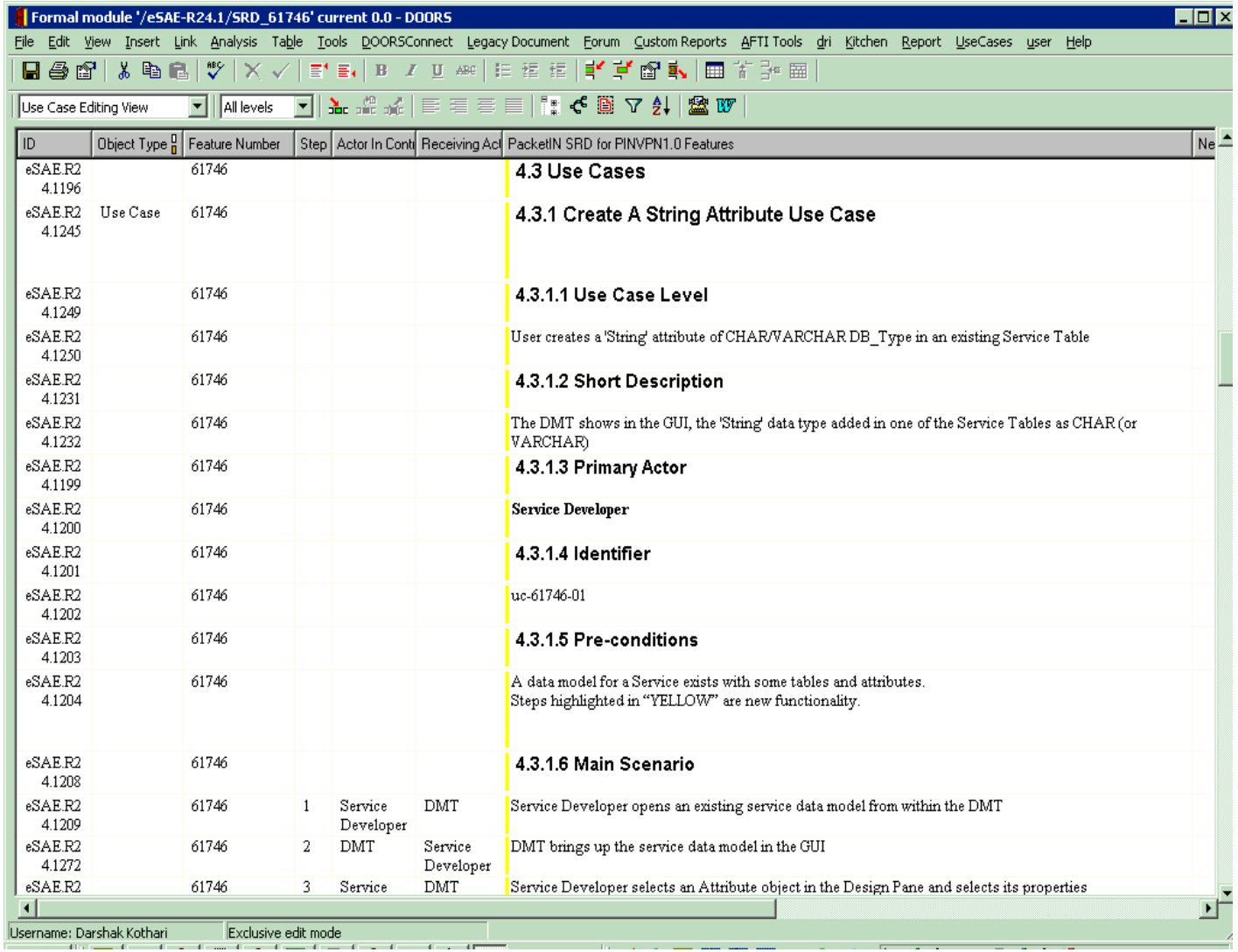


Matrice di tracciamento dei requisiti

Requirement Identifiers	Reqs Tested	REQ1 UC 1.1	REQ1 UC 1.2	REQ1 UC 1.3	REQ1 UC 2.1	REQ1 UC 2.2	REQ1 UC 2.3.1	REQ1 UC 2.3.2	REQ1 UC 2.3.3	REQ1 UC 2.4	REQ1 UC 3.1	REQ1 UC 3.2	REQ1 TECH 1.1	REQ1 TECH 1.2	REQ1 TECH 1.3
Test Cases	321	3	2	3	1	1	1	1	1	1	2	3	1	1	1
Tested Implicitly	77														
1.1.1	1	x													
1.1.2	2		x	x											
1.1.3	2	x											x		
1.1.4	1			x											
1.1.5	2	x												x	
1.1.6	1		x												
1.1.7	1			x											
1.2.1	2				x		x								
1.2.2	2					x		x							
1.2.3	2								x	x					
1.3.1	1										x				
1.3.2	1										x				
1.3.3	1											x			
1.3.4	1											x			
1.3.5	1											x			
etc....															
5.6.2	1														x

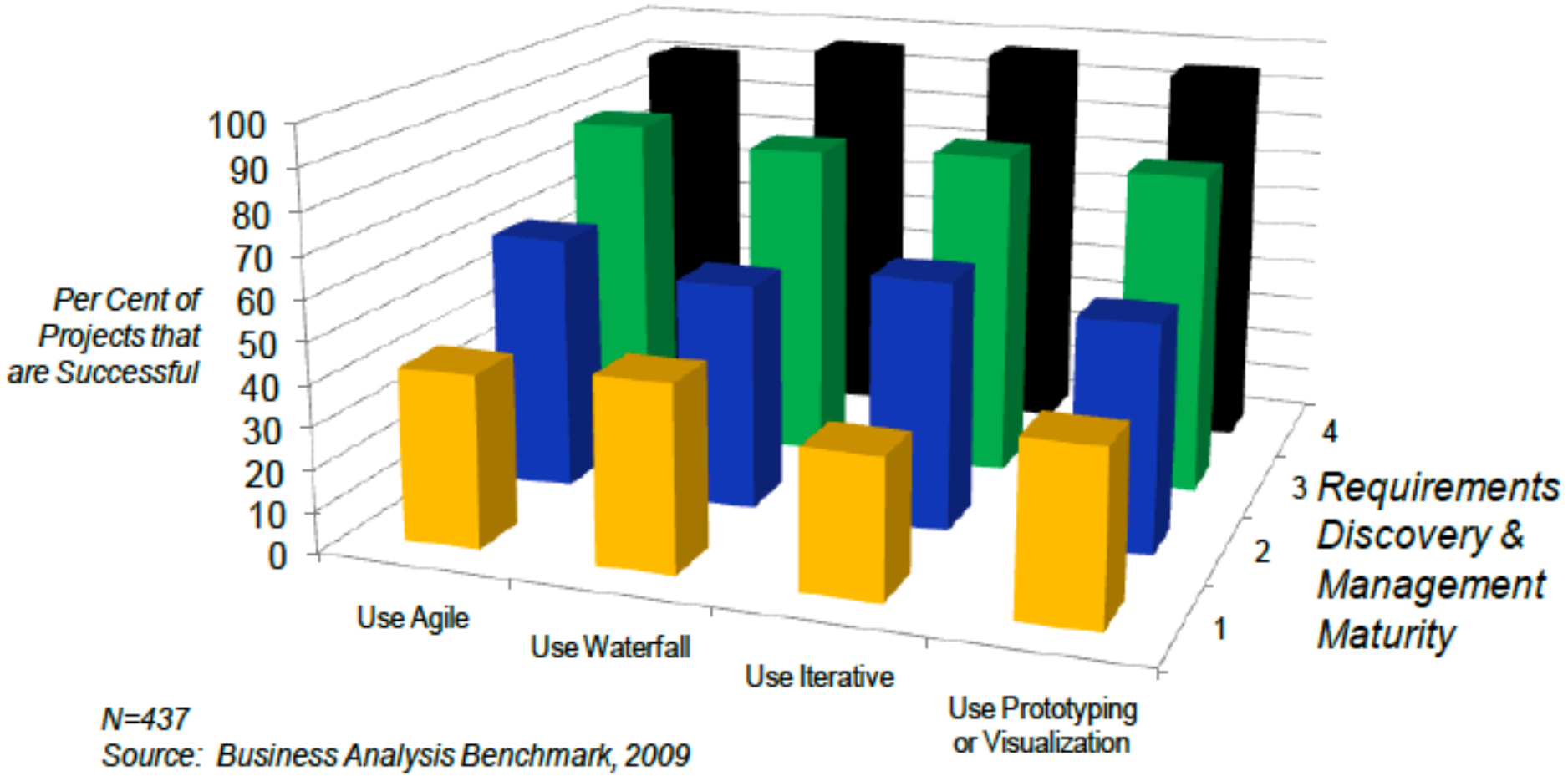
Fonte: Wikipedia

Use Cases in DOORS

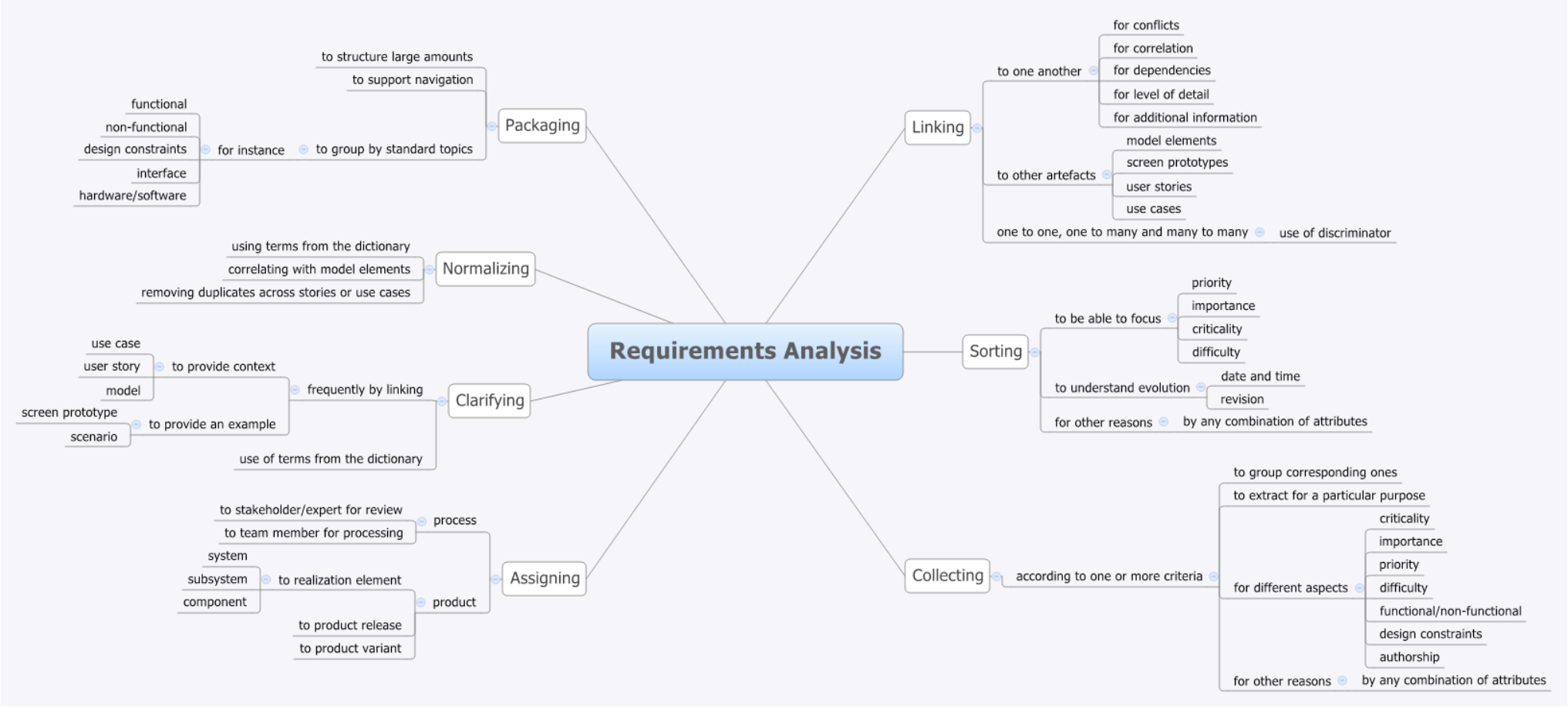


ID	Object Type	Feature Number	Step	Actor In Cont	Receiving Act	PacketIN SRD for PINVPN1.0 Features	Ne
eSAE.R2 4.1196		61746				**4.3 Use Cases**	
eSAE.R2 4.1245	Use Case	61746				**4.3.1 Create A String Attribute Use Case**	
eSAE.R2 4.1249		61746				**4.3.1.1 Use Case Level**	
eSAE.R2 4.1250		61746				User creates a 'String' attribute of CHAR/VARCHAR DB_Type in an existing Service Table	
eSAE.R2 4.1231		61746				**4.3.1.2 Short Description**	
eSAE.R2 4.1232		61746				The DMT shows in the GUI, the 'String' data type added in one of the Service Tables as CHAR (or VARCHAR)	
eSAE.R2 4.1199		61746				**4.3.1.3 Primary Actor**	
eSAE.R2 4.1200		61746				**Service Developer**	
eSAE.R2 4.1201		61746				**4.3.1.4 Identifier**	
eSAE.R2 4.1202		61746				uc-61746-01	
eSAE.R2 4.1203		61746				**4.3.1.5 Pre-conditions**	
eSAE.R2 4.1204		61746				A data model for a Service exists with some tables and attributes. Steps highlighted in "YELLOW" are new functionality.	
eSAE.R2 4.1208		61746				**4.3.1.6 Main Scenario**	
eSAE.R2 4.1209		61746	1	Service Developer	DMT	Service Developer opens an existing service data model from within the DMT	
eSAE.R2 4.1272		61746	2	DMT	Service Developer	DMT brings up the service data model in the GUI	
eSAE.R2		61746	3	Service	DMT	Service Developer selects an Attribute object in the Design Pane and selects its properties	

Importanza del processo di gestione dei requisiti



Mappa concettuale: l’analisi dei requisiti



da: Open Requirement Management Framework

Sommario

- Le feature di un prodotto sono di solito poche decine, i requisiti possono essere centinaia o anche migliaia
- Le feature danno una descrizione ad alto livello delle responsabilità e dei servizi che offre un sistema per rispondere ai bisogni degli stakeholder
- I requisiti servono per definire le feature; la distinzione più importante è tra funzionali e non
- I casi d'uso sono una tecnica per analizzare e classificare i requisiti funzionali
- Le user stories sono la tecnica agile per descrivere i requisiti

Feature

- Quando qualcuno costruisce un prodotto deve definire le *feature* del prodotto
- Le *feature* saranno la base per confrontare il prodotto con altri prodotti simili o per decidere come il prodotto va modificato per soddisfare nuovi requisiti

Requisiti

- Quando qualcuno chiede a qualcun altro di creare un sistema software occorre stabilire i *requisiti* di ciò che verrà costruito
- I requisiti saranno la base del testing e di ogni verifica

Domande di autotest

- Cos’è una *user story*?
- Cos'è un *caso d'uso*?
- Cos'è uno *scenario*?
- Qual’è la differenza tra “*extend*” e “*include*”?
- Cos’è una *feature* di un prodotto sw?
- Che cos’è un requisito di sistema?
- Quale struttura ha il documento della Specifica dei Requisiti Software (SRS)?
- Cos'è un requisito *non funzionale*?
- Perché è importante “*tracciare*” i requisiti?

Letture raccomandate

Zielczynski, *Traceability from Use Cases to Test Cases*, 2006
`www.ibm.com/developerworks/rational/library/04/r-3217/`

K. Wiegers, Automating Requirements Management, 1999
`www.processimpact.com/articles/rm_tools.html`

Riferimenti

- Capitolo 2 del SWEBOK. “Requirements Engineering”
- IEEE830-1998 *Recommended Practice for Sw Requirements Specifications*
- Beatty & Chen, *Visual Modeling for Software Requirements*, Microsoft Press, 2012
- Berenback et al, *Software & Systems Requirement Engineering in Practice*, McGrawHill, 2009
- Cockburn, *Writing Effective Use Cases*, Addison-Wesley, 2001
- Cohn, *User stories applied*, Addison-Wesley 2004
- Leffingwell and Widrig, *Managing Software Requirements: A Use Case Approach*, Addison-Wesley, 2003
- Wynne, The Cucumber Book, 2012

docs.ludost.net/PragmaticProgramming/The%20Cucumber%20Book%20Behaviour-Driven%20Development%20for%20Testers%20and%20Developers.pdf

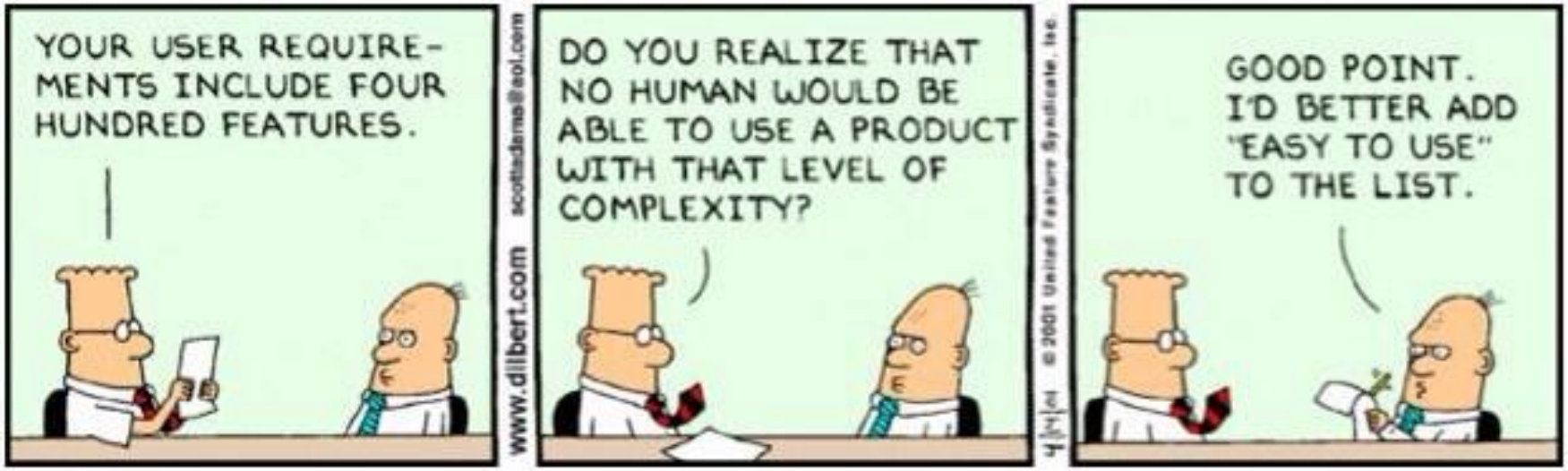
Siti utili

- github.com/cucumber/cucumber/wiki/Gherkin
- Cucumber cukes.info/
- www.telelogic.com/corp/products/doors/
- sourceforge.net/projects/eclipsesrs/
- sourceforge.net/projects/osrmt
- reqtracer.sourceforge.net
- www.flonatel.de/projekte/rmtoo/
- www.volere.co.uk/tools.htm
- www.volere.co.uk/template.htm
- www.analysttool.com/
- www.visual-paradigm.com/VPGallery/usecase/index.html
- www.workspace.com/workspace/Requirements-Management-Software.html

Pubblicazioni di ricerca sull'Ingegneria dei requisiti

- Int. Conf. on Requirements Engineering
- Requirements Engineering Journal

Domande?


YOUR USER REQUIREMENTS INCLUDE FOUR HUNDRED FEATURES.
DO YOU REALIZE THAT NO HUMAN WOULD BE ABLE TO USE A PRODUCT WITH THAT LEVEL OF COMPLEXITY?
GOOD POINT. I'D BETTER ADD "EASY TO USE" TO THE LIST.
www.dilbert.com
scottadams@aol.com
© 2001 United Feature Syndicate, Inc.