F11018: INGEGNERIA DEL SOFTWARE

Requirements Engineering

Lecturer: Prof. Henry Muccini
Università degli Studi dell'Aquila

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Università degli Studi dell'Aquila


1986 | 2024
FORMATICA FESTA
1986 | 2024
Corso di Laurea in Informatica
Università degli Studi dell'Aquila
21 22
OTTOBRE 2024
CENTRO CONGRESSI "LUIGI ZORDAN"
L'AQUILA
INGRESSO GRATUITO
Stiamo organizzando "Informatica In Festa: Software Cultura e Società", un evento aperto a tutti per scoprire l'importanza dell'informatica nella vita quotidiana. Esperti di fama discuteranno temi chiave per comprendere come la tecnologia influenzi la nostra società, il nostro modo di vivere e la nostra cultura. Un'occasione per guardare al futuro e capire insieme quali sfide e opportunità ci aspettano. Unisciti a noi per esplorare il mondo digitale in modo semplice e accessibile.
IN PROGRAMMA
L'INFORMATICA DALL'OLIVETTI ALL'INTELLIGENZA ARTIFICIALE: MODIFICHE, EVOLUZIONI E COMPETENZE CHE NON FINIRANNO MAI DI STUPIRE
IL LAVORO SMART IN UN MONDO DIGITALE: QUALE SOCIETÀ CI ASPETTA?
L'ETICA AL TEMPO DELL'INTELLIGENZA ARTIFICIALE
NAVIGARE LA TRASFORMAZIONE DIGITALE
INSEGNARE ED IMPARARE L'INFORMATICA AI TEMPI DELL'AI GENERATIVA
SPEAKERS
DARIO NIZZA _SME Development Director Adecco
MASSIMILIANO MONETTI_Presidente BorghiIN e delegato nazionale settore Cooperative di Comunità Confcooperative Habitat
PAOLA INVERARDI _Rettrice del Gran Sasso Science Institute
MASSIMO DI VIRGILIO_Presidente FIDAINFORM e CDTI Club Dirigenti Tecnologie dell'Informazione di Roma
PASQUALE LOVINO_Digital Transition Manager Mylia
ENZO DI NATALE_ Sindaco di Aielli
VINCENZO DI NICOLA_Presidente /ogita ETS
ENRICO NARDELLI_Università di "Roma Torvergata"
GIAMMARIA DE PAULIS_Imprenditore, Divulgatore Scientifico, Esperto in Comunicazione Digitale
MATTEO TESSI E ALESSANDRO ARELLA_Rete Ferroviaria Italiana

Copyright Notice

Some of the slides presented in this lecture comes from the textbook, some others from Chapter 4 of the Sommerville software engineering book

Henry Muccini

Text Book Reading

Chapter 4

«Object-Oriented Software Engineering – using UML, Patterns, and Java», by Bernd Bruegge and Allen H. Dutoit

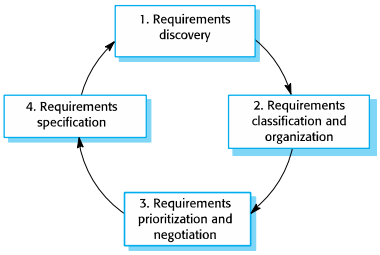
Chapter 4

«Software Engineering 10th Edition», by Ian Sommerville

http://iansommerville.com/software-engineering-book/

Topics covered

1. Functional and non-functional requirements
   precision, completeness, consistency
2. Requirements engineering processes
3. Requirements Discovery
4. Requirements classification
5. Requirements prioritization
6. Requirements specification



1-2. REQUIREMENT ENGINEERING

Topics covered

1.Functional and non-functional requirements
2.Requirements engineering processes
3.Requirements Discovery
4.Requirements classification
5.Requirements prioritization
6.Requirements specification

Requirements engineering

The process of establishing the services that a customer requires from a system, the qualities associated to the system and its services, and the constraints under which it operates and is developed.

Requirements engineering

- Services = requisiti funzionali
- Qualities = requisiti non funzionali
- Constraints = vincoli di sistema

What is a requirement?

- high-level abstract statement of a service or of a system constraint
- detailed mathematical functional specification.
- define the system **boundaries**

What is inside, what is outside the system?

Requirements may serve a dual function

- May be the basis for a bid for a contract - therefore must be open to interpretation;
- May be the basis for the contract itself - therefore must be defined in detail;

Functional and non-functional requirements

Functional requirements

- Statements of services the system should provide, how the system should react to particular inputs and how the system should behave in particular situations.
- May state what the system should not do.

Non-functional requirements

- Constraints on the services or functions offered by the system such as timing constraints, constraints on the development process, standards, etc.
- Often apply to the system as a whole rather than individual features or services.

Functional requirements

- Describe functionality or system services.
- Depend on the type of software, expected users and the type of system where the software is used.
- Functional user requirements may be high-level statements of what the system should do.
- Functional system requirements should describe the system services in detail.

Requirements imprecision

Problems arise when functional requirements are not precisely stated.

Ambiguous requirements may be interpreted in different ways by developers and users.

Consider the term 'search' in requirement 1

- User intention – search for a patient name across all appointments in all clinics;
- Developer interpretation – search for a patient name in an individual clinic. User chooses clinic then search.

Requirements completeness and consistency

In principle, requirements should be both complete and consistent.

Complete

- They should include descriptions of all facilities required.

Consistent

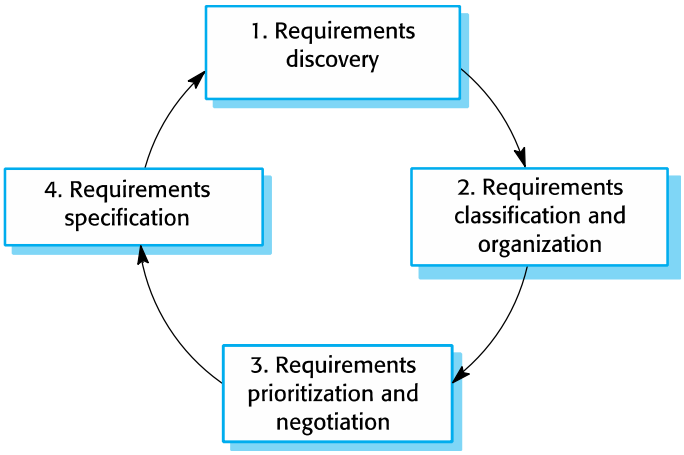
- There should be no conflicts or contradictions in the descriptions of the system facilities.

REQUIREMENTS ELICITATION

Requirements Elicitation

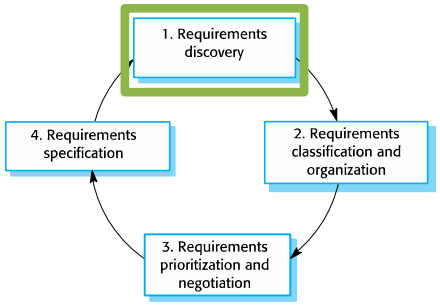
Involves technical staff working with customers to find out about the application domain, the services that the system should provide and the system's operational constraints.

The requirements elicitation and analysis process



Problems of requirements elicitation

- Stakeholders **don't know what they really want**.
- Stakeholders express requirements in their own terms.
- **Different stakeholders may have conflicting requirements**.
- Organisational and political factors may influence the system requirements.
- The requirements **change** during the analysis process. New stakeholders may emerge and the business environment may change.



3. REQUIREMENTS DISCOVERY

Techniques to DISCOVER Requirements

Bridging the gap between end user and developer:

- **Questionnaires & Interviews:** Asking the end user a list of pre-selected questions
- **Ethnography:** Observing end users in their operational environment
- **User Stories &Scenarios:** Describe the use of the system as a series of interactions between a concrete end user and the system
- **Use cases:** Abstractions that describe a class of scenarios.

discussion

Scenarios

- Scenarios and user stories are real-life examples of how a system can be used
- A synthetic description of an event or series of actions and events.
- A scenario can include text, video, pictures and story boards. It usually also contains details about the work place, social situations and resource constraints.

Example

- Customers can search for books on the Murray.com website, either by author name, or words in the title. A list of all matching books is returned to the customer. A customer does not need to be logged-in in order to search.
- The system records all the customers of the Murray.com who have ever logged in. A customer may be an individual customer or a business customer.
- Each customer has a username and password. Business customers may have several usernames and passwords, corresponding to different divisions within the business.
- When a customer has selected a book to buy at the Murray.com website. The system prompts for the customer's username and password. The customer enters these details. The system verifies the customer's identity and retrieves the customer's name and address, then prompts for credit card details. The customer enters these details. The system checks the credit card details. The system shows the customer the book and delivery price. The customer confirms the transaction.
- The system records all books available at Murray.com. For each book, the author, title and ISBN number are recorded. The number of each book in stock is also stored, along with the number on order by customers and the number on order from publishers. Books may be temporarily unavailable

Scenario-based Design

- Focuses on concrete descriptions and particular instances, not abstract generic ideas
- It is work driven not technology driven
- It is open-ended, it does not try to be complete
- It is informal, not formal and rigorous
- Is about envisioned outcomes, not about specified outcomes.

Scenarios: should include

Scenarios should include

- A description of the starting situation;
- A description of the normal flow of events;
- A description of what can go wrong;
- Information about other concurrent activities;
- A description of the state when the scenario finishes.

Nome Scenario (con ID)	**SC-4**: Utente che usufruisce di un servizio
Attori Coinvolti	**A-4**
Precondizione	L'utente si reca presso il relativo servizio (es. palestra).
Servizi Associati (ID)	**RF-5**
Flusso di eventi	Per poter entrare dovrà passare per una porta con Smart Lock, la quale per essere sbloccata dovrà acquisire la carta elettronica dell'utente attraverso il lettore RFID integrato. All'uscita dal luogo del servizio l'utente dovrà nuovamente passare per una porta di uscita, la quale verrà sempre sbloccata con la lettura della carta elettronica.
Cosa Può Andare Storto	1. L'utente non ha acquistato tale servizio. In tal caso si procede come descritto dallo scenario **SC-2.** 2. La carta fornita al check-in è difettosa o non è stata programmata correttamente.
Criticità	
Altre Attività	

A description of the starting situation

System stakeholders

Any person or organization who is affected by the system in some way and so who has a legitimate interest

Stakeholder types

- End users
- System managers
- System owners
- External stakeholders

Heuristics for finding scenarios

Ask yourself or the client the following questions:

- What are the primary tasks that the system needs to perform?
- What data will the actor create, store, change, remove or add in the system?
- What external changes does the system need to know about?
- What changes or events will the actor of the system need to be informed about?

However, don't rely on questions *and* questionnaires alone

Insist on task observation if the system already exists (interface engineering or reengineering)

- Ask to speak to the end user, not just to the client
- Expect resistance and try to overcome it.

Observations

Concrete scenario

- Describes a single instance of reporting a fire incident.
- Does not describe all possible situations in which a fire can be reported.

Participating actors

- Luca: the voter, Francesca: the official, Maura: the voter

After the scenarios are formulated

Find all the use cases in the scenario that specify all instances of how to report a fire

- Example:"Report Emergency" in the first paragraph of the scenario is a candidate for a use case

Describe each of these use cases in more detail

- Participating actors
- Describe the entry condition
- Describe the flow of events
- Describe the exit condition
- Describe exceptions
- Describe nonfunctional requirements

Functional Modeling (see next lecture)

Dashboard di Monitoraggio e Gestione delle Alluvioni

Obiettivo:
Realizzare una dashboard che permetta di monitorare in tempo reale dati provenienti da sensori distribuiti in zone a rischio alluvione all'interno di una città o di una regione. Questi sensori rileveranno parametri chiave come livello dell'acqua, intensità della pioggia, saturazione del suolo e velocità del vento. Ogni sensore avrà un codice identificativo univoco e sarà associato a una zona specifica soggetta a rischio idrogeologico.

Descrizione del sistema:
I sensori, distribuiti in diverse aree geografiche come fiumi, canali, bacini di raccolta e aree urbane a rischio, raccoglieranno periodicamente i seguenti parametri:

- **Livello dell'acqua**
- **Velocità di flusso dei fiumi**
- **Pioggia cumulativa**
- **Saturazione del terreno**
- **Velocità e direzione del vento**

Ogni sensore invierà periodicamente i dati al sistema centrale insieme allo stato di funzionamento del sensore stesso (0-1). La dashboard mostrerà i dati raccolti e informerà i gestori di eventuali superamenti delle soglie critiche predefinite, come ad esempio il superamento del livello di allerta di un fiume o l'intensità delle piogge.

Funzionalità base:

- Visualizzazione dei dati
- Possibilità di selezionare un'area geografica specifica e monitorare i sensori associati a tale area.
- Notifica di malfunzionamento dei sensori
- Invio di allarmi prioritari in caso di pericolo imminente, come il rischio di esondazione in corso.

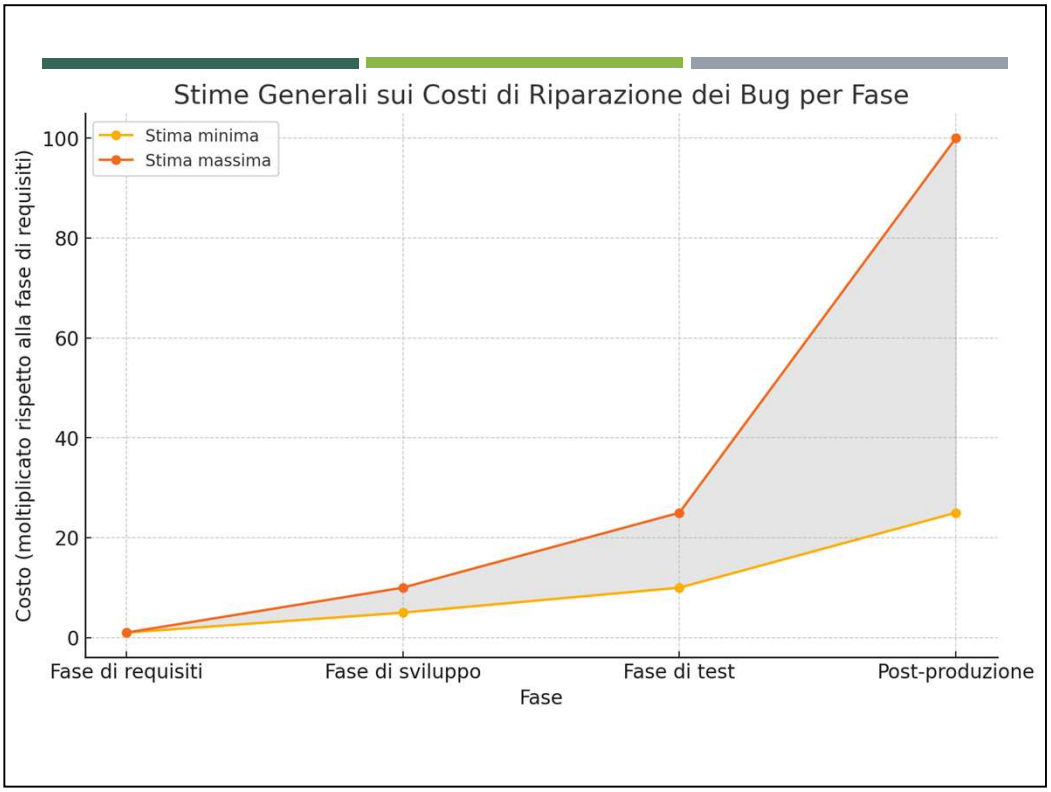
IN CLASSE

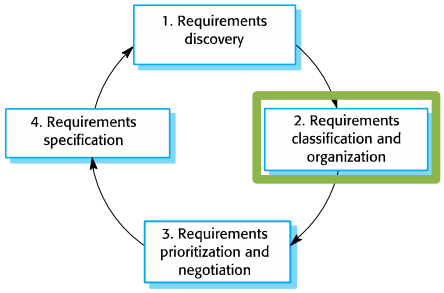
Requirements Discovery: Difficulties and Challenges

My Experience:

challenges

boundaries


Stime Generali sui Costi di Riparazione dei Bug per Fase
Stima minima
Stima massima
Costo (moltiplicato rispetto alla fase di requisiti)
100
80
60
40
20
0
Fase di requisiti
Fase di sviluppo
Fase di test
Post-produzione
Fase

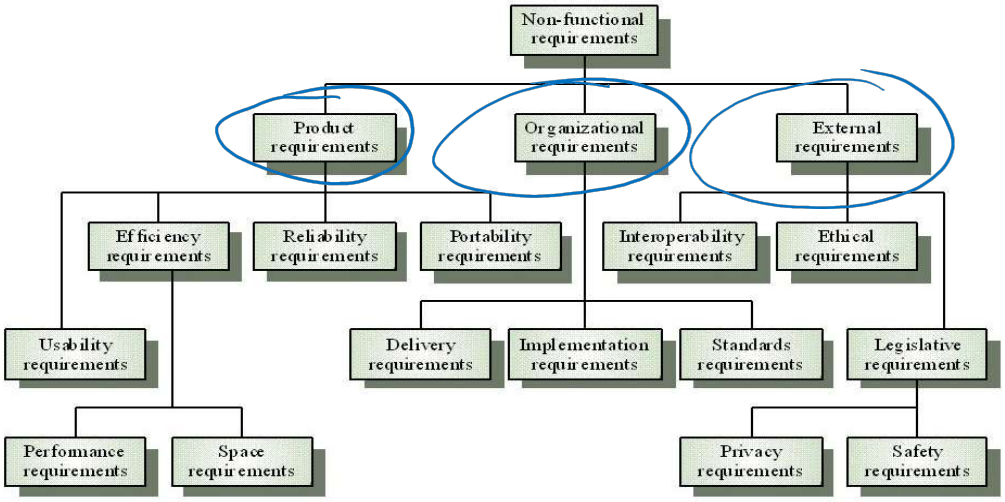


REQUIREMENTS CLASSIFICATION AND ORGANIZATION

CLASSIFICAZIONE DEI REQUISITI

- Funzionali
- Non funzionali

UNA TASSONOMIA DI REQUISITI NON FUNZIONALI



Example

- UFFIZI spreedsheet

Dashboard di Monitoraggio e Gestione delle Alluvioni

Obiettivo:
Realizzare una dashboard che permetta di monitorare in tempo reale dati provenienti da sensori distribuiti in zone a rischio alluvione all'interno di una città o di una regione. Questi sensori rileveranno parametri chiave come livello dell'acqua, intensità della pioggia, saturazione del suolo e velocità del vento. Ogni sensore avrà un codice identificativo univoco e sarà associato a una zona specifica soggetta a rischio idrogeologico.

Descrizione del sistema:
I sensori, distribuiti in diverse aree geografiche come fiumi, canali, bacini di raccolta e aree urbane a rischio, raccoglieranno periodicamente i seguenti parametri:

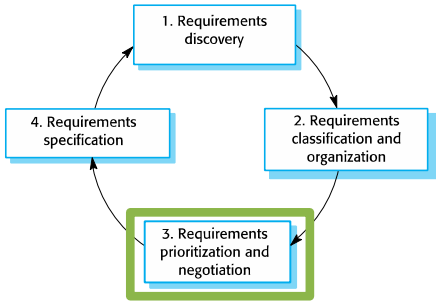
- **Livello dell'acqua**
- **Velocità di flusso dei fiumi**
- **Pioggia cumulativa**
- **Saturazione del terreno**
- **Velocità e direzione del vento**

Ogni sensore invierà periodicamente i dati al sistema centrale insieme allo stato di funzionamento del sensore stesso (0-1). La dashboard mostrerà i dati raccolti e informerà i gestori di eventuali superamenti delle soglie critiche predefinite, come ad esempio il superamento del livello di allerta di un fiume o l'intensità delle piogge.

Funzionalità base:

- Visualizzazione dei dati
- Possibilità di selezionare un'area geografica specifica e monitorare i sensori associati a tale area.
- Notifica di malfunzionamento dei sensori
- Invio di allarmi prioritari in caso di pericolo imminente, come il rischio di esondazione in corso.

IN CLASSE



REQUIREMENTS PRIORITIZATION AND NEGOTIATION

Prioritizing requirements

High priority

- Addressed during analysis, design, and implementation
- A high-priority feature must be demonstrated

Medium priority

- Addressed during analysis and design
- Usually demonstrated in the second iteration

Low priority

- Addressed only during analysis
- Illustrates how the system is going to be used in the future with not yet available technology

High priority ("Core requirements")

Must be addressed during analysis, design, and implementation.

A high-priority feature must be demonstrated successfully during client acceptance.

Medium priority ("Optional requirements")

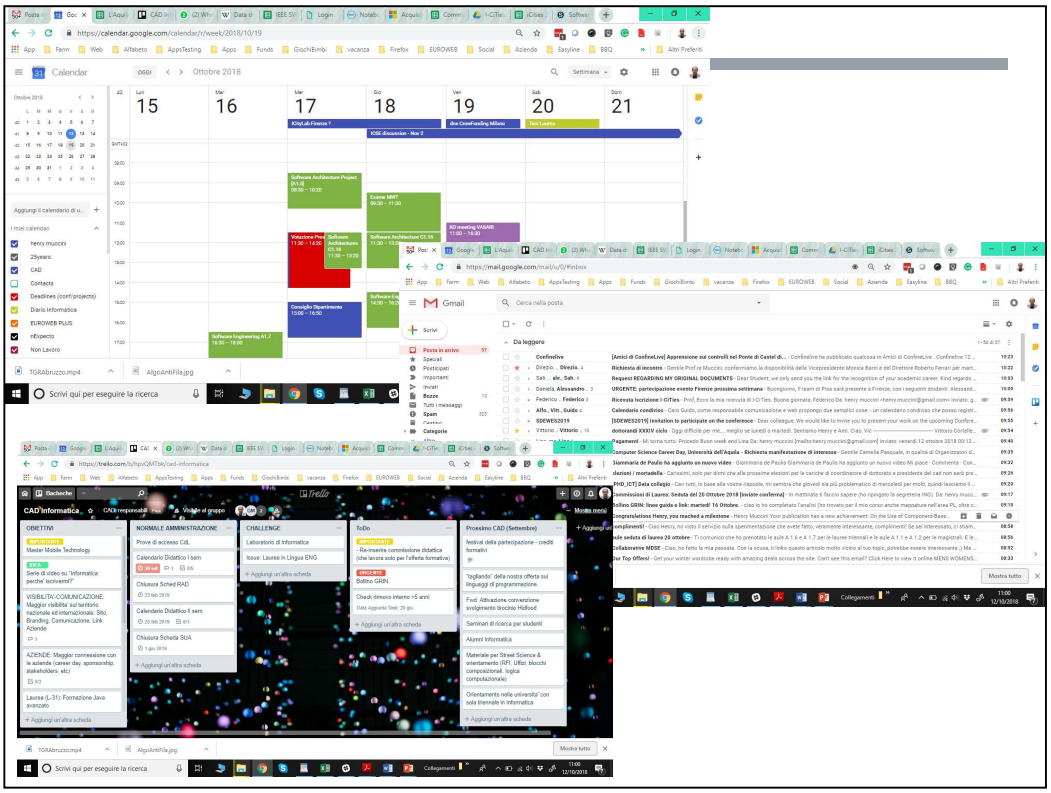
Must be addressed during analysis and design.

Usually implemented and demonstrated in the second iteration of the system development.

Low priority ("Fancy requirements")

Must be addressed during analysis ("very visionary scenarios").

Illustrates how the system is going to be used in the future if not yet available technology enablers are available


Calendar
Gmail
CAD Informatica
Trello

Dashboard di Monitoraggio e Gestione delle Alluvioni

Obiettivo:
Realizzare una dashboard che permetta di monitorare in tempo reale dati provenienti da sensori distribuiti in zone a rischio alluvione all'interno di una città o di una regione. Questi sensori rileveranno parametri chiave come livello dell'acqua, intensità della pioggia, saturazione del suolo e velocità del vento. Ogni sensore avrà un codice identificativo univoco e sarà associato a una zona specifica soggetta a rischio idrogeologico.

Descrizione del sistema:
I sensori, distribuiti in diverse aree geografiche come fiumi, canali, bacini di raccolta e aree urbane a rischio, raccoglieranno periodicamente i seguenti parametri:

- **Livello dell'acqua**
- **Velocità di flusso dei fiumi**
- **Pioggia cumulativa**
- **Saturazione del terreno**
- **Velocità e direzione del vento**

Ogni sensore invierà periodicamente i dati al sistema centrale insieme allo stato di funzionamento del sensore stesso (0-1). La dashboard mostrerà i dati raccolti e informerà i gestori di eventuali superamenti delle soglie critiche predefinite, come ad esempio il superamento del livello di allerta di un fiume o l'intensità delle piogge.

Funzionalità base:

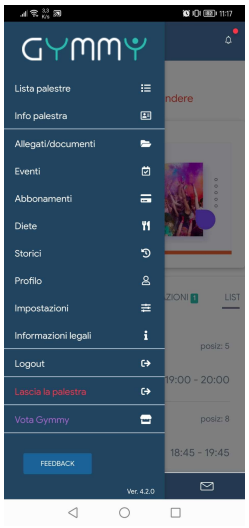
- Visualizzazione dei dati
- Possibilità di selezionare un'area geografica specifica e monitorare i sensori associati a tale area.
- Notifica di malfunzionamento dei sensori
- Invio di allarmi prioritari in caso di pericolo imminente, come il rischio di esondazione in corso.

IN CLASSE

PRIMA DI ANDARE OLTRE…

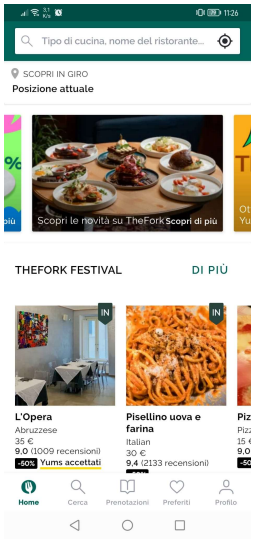
Bring

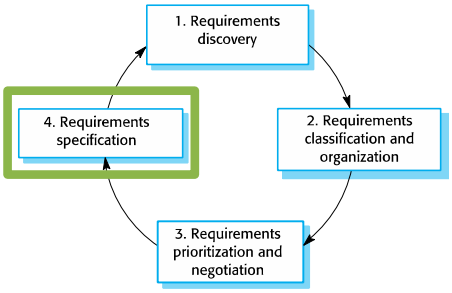
Gymmy



FIRENZE Guida Biglietti & Tour

The Fork





REQUIREMENTS SPECIFICATION

Ways of writing a system requirements specification

Notation	Description
Natural language	The requirements are written using numbered sentences in natural language. Each sentence should express one requirement.
Structured natural language	The requirements are written in natural language on a standard form or template. Each field provides information about an aspect of the requirement.
Design description languages	This approach uses a language like a programming language, but with more abstract features to specify the requirements by defining an operational model of the system. This approach is now rarely used although it can be useful for interface specifications.
Graphical notations	Graphical models, supplemented by text annotations, are used to define the functional requirements for the system; UML use case and sequence diagrams are commonly used.
Mathematical specifications	These notations are based on mathematical concepts such as finite-state machines or sets. Although these unambiguous specifications can reduce the ambiguity in a requirements document, most customers don't understand a formal specification. They cannot check that it represents what they want and are reluctant to accept it as a system contract

Problems with natural language

Lack of clarity

- Precision is difficult without making the document difficult to read.

Requirements confusion

- Functional and non-functional requirements tend to be mixed-up.

Requirements amalgamation

- Several different requirements may be expressed together.

Example requirements for the insulin pump software system

3.2 The system shall measure the blood sugar and deliver insulin, if required, every 10 minutes. *(Changes in blood sugar are relatively slow so more frequent measurement is unnecessary; less frequent measurement could lead to unnecessarily high sugar levels.)*

3.6 The system shall run a self-test routine every minute with the conditions to be tested and the associated actions defined in Table 1. *(A self-test routine can discover hardware and software problems and alert the user to the fact the normal operation may be impossible.)*

Structured specifications

- An approach to writing requirements where the freedom of the requirements writer is limited and requirements are written in a standard way.

- This works well for some types of requirements e.g. requirements for embedded control system but is sometimes too rigid for writing business system requirements.

A structured specification of a requirement for an insulin pump

Insulin Pump/Control Software/SRS/3.3.2

Function Compute insulin dose: safe sugar level.

Description

Computes the dose of insulin to be delivered when the current measured sugar level is in the safe zone between 3 and 7 units.

Inputs Current sugar reading (r2); the previous two readings (r0 and r1).

Source Current sugar reading from sensor. Other readings from memory.

Outputs CompDose—the dose in insulin to be delivered.

Destination Main control loop.

A structured specification of a requirement for an insulin pump

Action	
CompDose is zero if the sugar level is stable or falling or if the level is increasing but the rate of increase is decreasing. If the level is increasing and the rate of increase is increasing, then CompDose is computed by dividing the difference between the current sugar level and the previous level by 4 and rounding the result. If the result, is rounded to zero then CompDose is set to the minimum dose that can be delivered.	
Requirements	
Two previous readings so that the rate of change of sugar level can be computed.	
Pre-condition	
The insulin reservoir contains at least the maximum allowed single dose of insulin.	
Post-condition	r0 is replaced by r1 then r1 is replaced by r2.
Side effects	None.

Tabular specification of computation for an insulin pump

Condition	Action
Sugar level falling (r2 < r1)	CompDose = 0
Sugar level stable (r2 = r1)	CompDose = 0
Sugar level increasing and rate of increase decreasing ((r2 – r1) < (r1 – r0))	CompDose = 0
Sugar level increasing and rate of increase stable or increasing ((r2 – r1) ≥ (r1 – r0))	CompDose = round ((r2 – r1)/4) If rounded result = 0 then CompDose = MinimumDose

TOOLS

Gran varieta':

- Documenti condivisi in GDrive/Dropbox/Teams etc.

- UML tools per Use Case Diagrams
- IBM Rational DOORS

- ReqSuite RM
- JIRA + Confluence
- Trello + Confluence
- GitHub, GitLab
- …

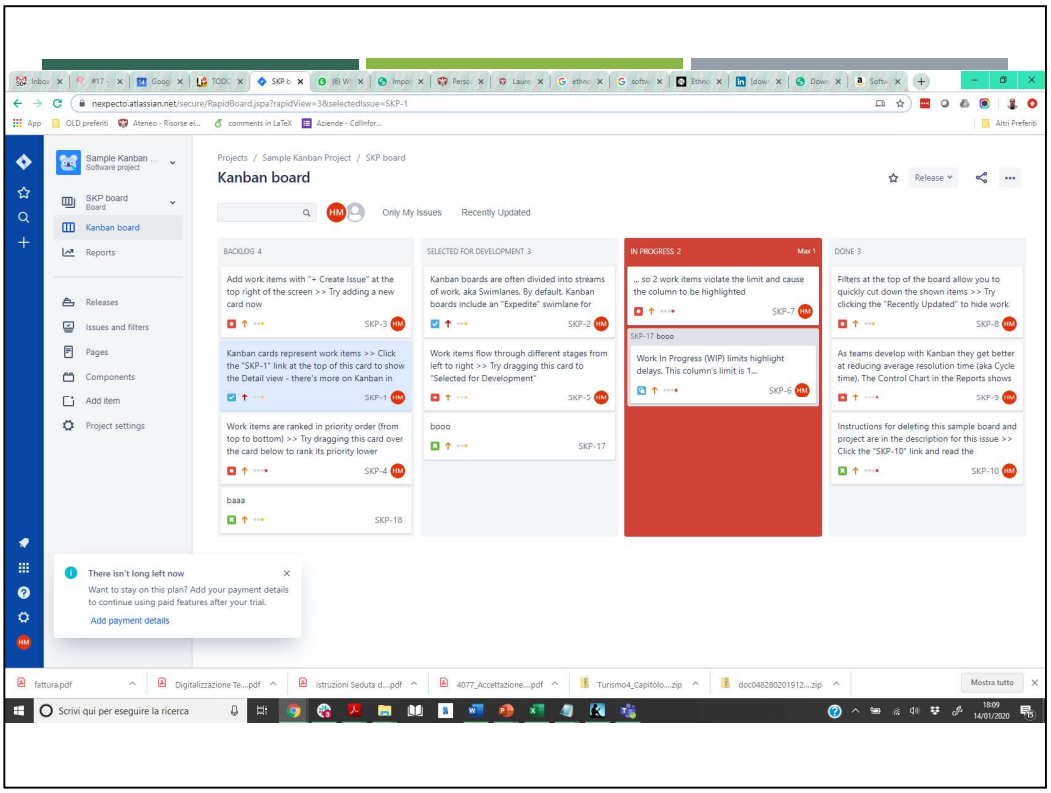
Come selezionarne uno appropriato?

SUGGERIMENTI PER LA SELEZIONE DI UN TOOL/AMBIENTE

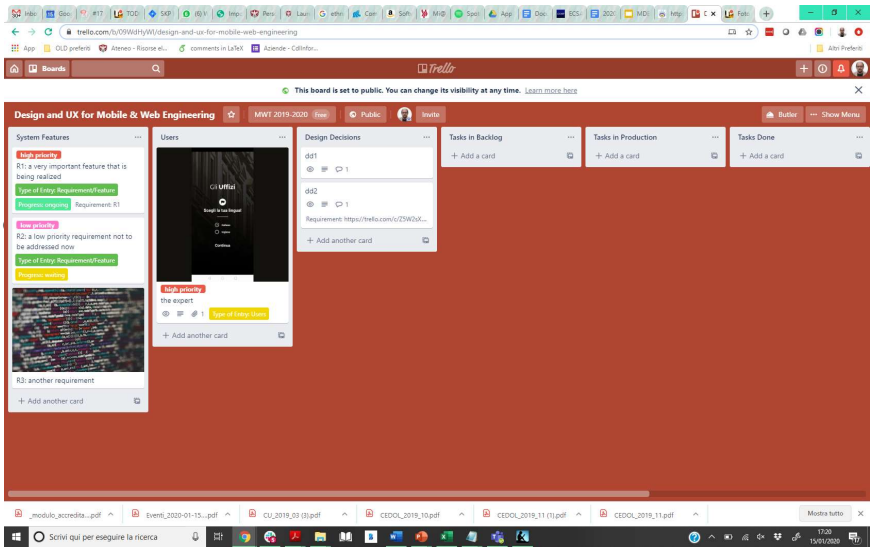
- **Ambienti integrati:**
- che permettano di creare «trace link» automatici tra specifica Req, task, issue trackers, design decisions, API, test, etc.

- **Ambienti collaborativi:**
- che permettano di lavorare in team in modo semplice ed efficace

- **Web-based:**
- che possano essere acceduti in modalità desktop/mobile, in qualsiasi momento


Sample Kanban ...
Software project
SKP board
Board
Kanban board
Reports
Releases
Issues and filters
Pages
Components
Add item
Project settings
Projects / Sample Kanban Project / SKP board
Kanban board
Release
Only My Issues
Recently Updated
BACKLOG 4
Add work items with "+ Create Issue" at the top right of the screen >> Try adding a new card now
SKP-3
Kanban cards represent work items >> Click the "SKP-1" link at the top of this card to show the Detail view - there's more on Kanban in
SKP-1
Work items are ranked in priority order (from top to bottom) >> Try dragging this card over the card below to rank its priority lower
SKP-4
baaa
SKP-18
SELECTED FOR DEVELOPMENT 3
Kanban boards are often divided into streams of work, aka Swimlanes. By default, Kanban boards include an "Expedite" swimlane for
SKP-2
Work items flow through different stages from left to right >> Try dragging this card to "Selected for Development"
SKP-5
booo
SKP-17
IN PROGRESS 2
Max 1
... so 2 work items violate the limit and cause the column to be highlighted
SKP-7
SKP-17 booo
Work In Progress (WIP) limits highlight delays. This column's limit is 1...
SKP-6
DONE 3
Filters at the top of the board allow you to quickly cut down the shown items >> Try clicking the "Recently Updated" to hide work
SKP-8
As teams develop with Kanban they get better at reducing average resolution time (aka Cycle time). The Control Chart in the Reports shows
SKP-9
Instructions for deleting this sample board and project are in the description for this issue >> Click the "SKP-10" link and read the
SKP-10
There isn't long left now
Want to stay on this plan? Add your payment details to continue using paid features after your trial.
Add payment details
Mostra tutto
Scrivi qui per eseguire la ricerca

TRELLO


Boards
Trello
This board is set to public. You can change its visibility at any time. Learn more here
Design and UX for Mobile & Web Engineering
MWT 2019-2020
Public
Invite
Butler
Show Menu
System Features
High priority
R1: a very important feature that is being realized
Type of Entry: Requirement/Feature
Progress: ongoing
Requirement: R1
Low priority
R2: a low priority requirement not to be addressed now
Type of Entry: Requirement/Feature
Progress: waiting
R3: another requirement
Add another card
Users
Gli Uffizi
High priority
the expert
Type of Entry: Users
Add another card
Design Decisions
dd1
dd2
Add another card
Tasks in Backlog
Add a card
Tasks in Production
Add a card
Tasks Done
Add a card
Mostra tutto
Scrivi qui per eseguire la ricerca

Esercizio per casa:

selezionate e studiate in dettaglio tre NFR, definite le relazioni tra di essi, ed applicateli ad una delle app descritte in precedenza

Standard ISO 25000 (http://iso25000.com/)

- ISO/IEC 25010: http://iso25000.com/index.php/en/iso-25000-standards/iso-25010
- ISO/IEC 25012: http://iso25000.com/index.php/en/iso-25000-standards/iso-25012

https://en.wikipedia.org/wiki/Non-functional_requirement

https://www.requirements.com/Glossary/NonFunctionalRequirements/tabid/91/Default.aspx

Textbook