

# Analisi dei requisiti

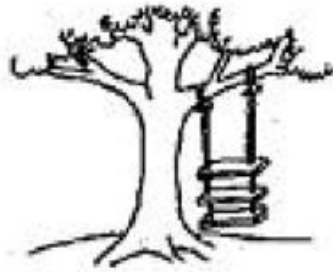
Roberta Gori, Laura Semini  
Corso di Ingegneria del Software  
Dipartimento di Informatica  
Università di Pisa

# Riassunto lezione precedente

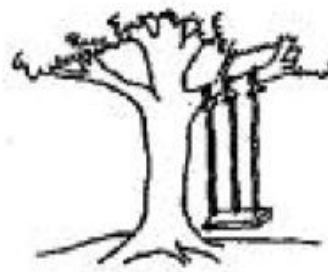
## Outline della lezione

- Lezione precedente:
  - Organizzazione del lavoro per realizzare un sistema sw
    - Processo sw
    - Modelli di ciclo di vita del sw
- Questa lezione
  - Studio di fattibilità
  - Analisi dei requisiti
    - Importanza dell'attività di analisi dei requisiti
    - Dominio
      - Comprensione e modellazione
    - Requisiti
      - Acquisizione e analisi

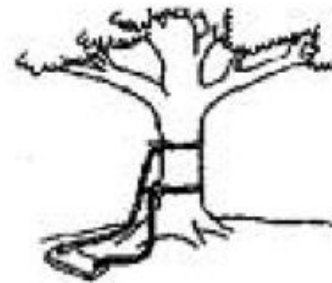
# L'altalena dei requisiti (una vignetta storica)



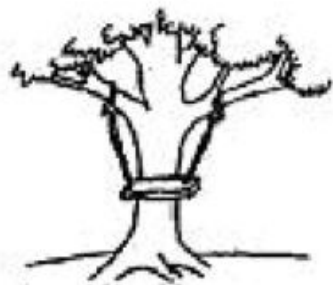
**As proposed  
by the project  
sponsor.**



**As specified  
in the project  
request.**



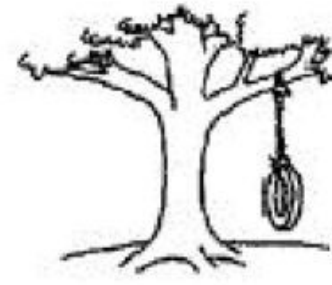
**As designed  
by the senior  
architect.**



**As produced  
by the  
engineers.**



**As installed at  
the user's  
site.**



**What the  
customer  
really wanted.**

# Studio di fattibilità

- Fase preliminare per stabilire l'opportunità o meno di realizzare il software
- Si basa su
  - Una descrizione sommaria del sistema software e delle necessità utente
  - Un'analisi per capire se ha senso realizzare il prodotto:
    - Cosa esiste già
    - Il prodotto venderà?
  - Un'analisi tecnica per capire se è realizzabile

# Studio di fattibilità

- Si basa sulla valutazione dei costi e dei benefici di una possibile attività di produzione
- Fattibilità tecnologica
  - Strumenti per la realizzazione (software, librerie, ...)
  - Soluzioni algoritmiche e architetturali
  - Hardware
  - Processo ( prototipazione, progetti esplorativi, ricerca..)
- Aspetti economici e di mercato
  - Confronto tra il mercato attuale e quello futuro
  - Costo della produzione, redditività dell'investimento

# Attività di analisi dei requisiti

- Studiare e definire il problema da risolvere
  - Per capire cosa deve essere realizzato
  - Per documentare cosa deve essere realizzato
  - Per negoziazioni committente/fornitore

# SLOGAN

cosa

non come

# Implicazioni economiche e di qualità (repetita iuvant)

- Le cause di abbandono secondo lo Standish Group
  1. Requisiti incompleti
  2. Scarso coinvolgimento degli utenti
  3. Mancanza di risorse
  4. Attese irrealistiche
  6. Modifiche a specifiche e requisiti
  - [.....]
  10. Ignoranza tecnologica



# Quanto "costa" correggere un errore nei requisiti [Boehm]

<b>FASE</b>	<b>COSTO</b>
<b>Analisi Requisiti</b>	<b>1</b>
<b>Progettazione</b>	<b>5</b>
<b>Codifica</b>	<b>10</b>
<b>Test di unità</b>	<b>20</b>
<b>Test di accettazione</b>	<b>50</b>
<b>Operazione</b>	<b>100</b>

# Prodotto dell'attività di analisi

- Documento e/o modello analitico di
  - Descrizione del dominio
  - Descrizione dei requisiti
- Opzionalmente anche
  - Manuale utente
  - Casi di test
- (spesso in parallelo)

# Dominio

- Dominio
  - Il campo di applicazione del prodotto
- Prima ancora di incontrare i committenti il team di analisti deve acquisire conoscenza del dominio di applicazione
  - Per poter porre le domande giuste domande: pensate al software per transazioni finanziarie
  - Per capire la differenza sottile che può esserci tra termini solo apparentemente sinonimi: pensate a supporto, montante, sostegno, trave in ingegneria civile

# Dominio

- Per acquisire conoscenza e definire il dominio:
  - si costruisce un **glossario**, il team di analisti lo costruisce mentre studia il dominio e poi si arricchisce via via che si incontrano nuovi termini
  - Si definiscono un **modello statico** e un **modello dinamico**
- La conoscenza del dominio evolve dopo incontro con committenti, utenti, ecc

# Requisito

- Requisito
  - Una condizione o una capacità necessaria a un utente per risolvere un problema [..]
  - Una condizione (capacità) che deve essere soddisfatta (posseduta) [..] da un sistema [..] per soddisfare un contratto [..]

[Glossario IEEE]

- In altre parole:
  - Una proprietà che deve essere garantita dal sistema per soddisfare una necessità di un utente
    - Funzionalità, qualità, ...

# Acquisire conoscenza su dominio e requisiti: Interviste

- I membri del team di analisti incontrano i membri dell'organizzazione del committente
- Si procede con delle interviste che possono essere
  - strutturate
  - non strutturate
- Entrambe presentano vantaggi e rischi
- Difficile condurre una buona intervista

# Acquisire conoscenza su dominio e requisiti: altre tecniche

- Questionari a risposte multiple a tutti i membri rilevanti dell'organizzazione del committente
- Costruzione di prototipi
- Osservazione di futuri utenti al lavoro
- Studio di documenti

# Acquisire e descrivere i requisiti (funzionali): Casi d'uso

- I casi d'uso sono un altro modo per acquisire i requisiti
- Il caso d'uso è un modo in cui un utente può usare il prodotto
- Si prospetta al committente insieme al risultato atteso e si aspettano i commenti
- I casi d'uso devono includere non solo la sequenza di eventi corretta ma anche comportamenti inattesi: **le eccezioni.**



# Categorie di requisiti

## Requisiti funzionali

- Descrivono le funzionalità che il sistema deve realizzare
  - In termini di:
    - azioni che il sistema deve compiere
    - come il sistema software reagisce a specifici tipi di input
    - come si comporta in situazioni particolari.
- Tradizionalmente i requisiti a cui è dato maggior valore
- Esempi
  1. Il sistema software deve fornire un appropriato visualizzatore per i documenti memorizzati
  2. L'utente deve essere in grado di effettuare ricerche sia sull'intero insieme di basi di dati che su un loro sottoinsieme
  3. Ad ogni nuovo ordine deve essere associato un identificatore unico (Order\_ID)

# Categorie di requisiti

## Requisiti non funzionali

- Aka requisiti di qualità
- Descrivono le proprietà del sistema software in relazione a determinati servizi o funzioni e possono anche essere relativi al processo:
  - caratteristiche di qualità quali:
    - efficienza, affidabilità, safety, usabilità, interfaccia, sicurezza, robustezza ecc.
  - caratteristiche del processo di sviluppo:
    - standard di processo, uso di ambienti CASE, linguaggi di programmazione, metodi di sviluppo, ecc.
  - caratteristiche esterne:
    - interoperabilità con sistemi di altre organizzazioni, vincoli legislativi, ecc.
  - requisiti fisici (hw)

# Esempi di requisiti non funzionali

1. Il tempo di risposta del sistema all'inserimento della password utente deve essere inferiore a 10 sec
2. I documenti di progetto (deliverable) devono essere conformi allo standard XYZ-ABC-12345
3. Il sistema software non deve rilasciare ai suoi operatori nessuna informazione personale relativa ai clienti, tranne nominativo e identificatore

# Requisiti funzionali vs requisiti non funzionali

- E' bene tenerli separati.
- Esempio:
  - Il sistema deve validare il pin inserito dal cliente entro 3 secondi



- Il sistema deve validare il pin inserito dal cliente
- La validazione deve essere completata entro 3 secondi

# Esercizio: Bancomat

Si consideri un sistema per il prelievo automatizzato di denaro contante (Bancomat). Per poter utilizzare il sistema l'utente deve possedere una tessera magnetica. Il sistema deve mettere a disposizione le funzioni di prelievo, saldo, estratto conto. Il sistema deve essere disponibile a persone portatori di Handicap, deve garantire un tempo di risposta inferiore al minuto, e deve essere sviluppato su architettura X86. Le operazioni di prelievo devono richiedere autenticazione tramite un codice segreto memorizzato sulla carta. Il sistema deve essere facilmente espandibile, e adattabile alle future esigenze bancarie.

# Approcci diversi che vedremo nel corso (non mutuamente esclusivi nella pratica)

- Basato su documenti in linguaggio naturale
  - Glossario
  - Specifica dei requisiti
    - funzionali
    - di qualità (non funzionali)
- Basato su linguaggi formali (modelli/grafici)
  - UML
    - Modello del dominio
    - Casi d'uso
      - Requisiti funzionali

# In questa lezione

- Approccio basato sulla produzione di documenti in linguaggio naturale
- Output
  - Documento dei requisiti
    - Comprende il glossario

# Documento dei requisiti

- Il documento dei requisiti è un contratto tra lo sviluppatore e il committente.
- Specifica cosa il prodotto deve fare e quali sono i vincoli che deve soddisfare
- Specifica anche in genere una deadline per la consegna del prodotto



# Documento dei requisiti in 5 passi

1. Acquisizione
2. Elaborazione
3. Negoziazione
4. Convalida
5. Gestione

# Acquisizione (già visto)

- Interviste
  - Strutturate o non strutturate
- Questionari scritti
  - Scelte multiple, ....
- Osservazione di futuri utenti al lavoro
- Studio di documenti
- Produzione prototipi
- Casi d'uso
- Con committenti, utenti, responsabili marketing

# Elaborazione

- Requisiti espansi e raffinati
- Definizione del documento dei requisiti
  - Basato sull'uso del linguaggio naturale
  - Descrizione del dominio

# Struttura del documento dei requisiti (1/2)

Proposta da Sommerville, e ispirata allo **standard IEEE/ANSI 830-1993** su *requirements documents*

- Introduzione
  - Perché il sistema è desiderabile e come si inquadra negli obiettivi piu' generali del cliente
- Glossario
  - I termini e i concetti tecnici usati
- *Definizione* dei Requisiti funzionali
  - I servizi richiesti
- *Definizione* dei Requisiti non funzionali
  - I vincoli operativi del sistema, e quelli sul processo di sviluppo
- Architettura
  - La strutturazione in sottosistemi (cui riferire i requisiti)

# Struttura del documento dei requisiti (2/2)

- Specifica di System and software requirements
  - Specifica dettagliata dei requisiti funzionali
- Modelli astratti del sistema
  - Modelli formali, o semi-formali: ciascuno illustra un solo punto di vista: controllo, dati, funzioni
- Evoluzione del sistema
  - Previsione di successivi cambiamenti (p. es. di HW, o di requisiti)
- Appendici
  - Individuazione ed eventuale descrizione della piattaforma hardware
  - Requisiti di DataBase
  - Manuale Utente, Piani di Test
- Indici
  - Lemmario: lista di termini, con puntatori ai requisiti che li usano

# Glossario

- Definizione dei termini chiave del dominio
- Strumento per comprendere e documentare il dominio
- Ma non solo:
  - Validazione dei requisiti

# Esempi di requisiti ben posti

## il <istema> deve <funzionalità>/<proprietà>

- Il sistema deve gestire tutti i registratori di cassa del negozio (non più di 20)
- Il sistema deve stampare la sintesi degli incassi della giornata
- A fine giornata il sistema deve elencare quali sono gli articoli da reintegrare in base alle vendite
- Il tempo di risposta a qualunque interrogazione deve essere inferiore a 3 secondi

# Documento dei requisiti in 5 passi

1. Acquisizione
2. Elaborazione
3. Convalida
4. Negoziazione
5. Gestione



# Convalida dei requisiti

Validazione di un documento già strutturato

- Deskcheck
  - Walkthrough
    - lettura sequenziale dei documenti
  - Ispezione
    - lettura “strutturata” dei documenti
    - Esempi:
      - tecnica del lemmario
      - Ricerca di rimozioni, distorsioni, generalizzazioni (Noam Chomsky 1975)
- Uso di strumenti di analisi del linguaggio naturale
- Prototipi

# Proprietà del documento dei requisiti

- Difetti da evitare
  - Omissioni
    - Mancata presenza di un requisito (incompletezza)
  - Inconsistenze
    - Contraddizione fra i vari requisiti o dei requisiti rispetto all'ambiente operativo
  - Ambiguità
    - Requisiti con significati multipli
  - Sinonimi e omonimi
    - Termini diversi con il medesimo significato e termini uguali con differenti significati
  - Presenza di dettagli tecnici
  - Ridondanza
    - Può esserci ridondanza, ma solo tra sezioni diverse.

# Incompletezze

- *Il sistema dovrà permettere agli utenti di fare delle ricerche per titolo, autore, o ISBN.*
- *Che cosa significa per i CD-ROM?*
  - Possono non avere un ISBN
  - Vale solo per i libri
  - Immaginate se realizzate il sistema con questo requisito senza considerare i CD-ROM.
- Naturalmente non possiamo scrivere requisiti universali, ma possiamo tentare di essere il più possibile completi.

# Non ambiguità

- Esempio: “Tutti i file sono controllati da un blocco di controllo”
- Interpretazioni:
  1. Un unico blocco di controllo controlla tutti i file
  2. Ogni file ha un suo specifico blocco di controllo
  3. Ogni file è controllato da uno specifico blocco di controllo; alcuni blocchi di controllo possono controllare più di un file
- Il linguaggio naturale è per sua natura ambiguo

# Convalida dei requisiti di esempio visti

1. Il sistema software deve fornire un **appropriato** visualizzatore per i documenti memorizzati
  - Appropriato è un termine vago che genera ambiguità
  - Per il reto requisito ben posto, in forma assertiva: Il sistema deve ...
2. L'utente deve essere in grado di effettuare ricerche sia sull'intero insieme di basi di dati che su **un loro sottoinsieme**
  - Un loro sottoinsieme è ambiguo: uno solo, e in caso quale? Tutti?
  - Per il reto requisito ben posto, in stile caso d'uso (con gli occhi dell'utente)
3. **Ad ogni nuovo ordine deve essere associato un identificatore unico (Order\_ID)**
  - Requisito di basso livello: introduce vincoli implementativi: è necessario ?

# Convalida dei requisiti: Tecnica del lemmario

- Termini del glossario, con puntatori ai requisiti che li nominano
- Facilita la ricerca di
  - Inconsistenze
  - Sinonimi
  - Omonimi
  - Ridondanze

# Convalida dei requisiti: strumenti di analisi lessicale e sintattica

- Analizzano la lista di requisiti in cerca di termini ambigui:
  - Il sistema **può** visualizzare le informazioni di tracciamento dell'ordine.
  - Dopo la consegna, il sistema deve consentire all'utente di inserire le proprie recensioni **o** valutazioni.
  - I prezzi dei prodotti in vendita sono espressi nella valuta **appropriata**

# Spazio per migliorare???

- Il sistema deve gestire tutti i registratori di cassa del negozio (non più di 20)
- Il sistema deve stampare la sintesi degli incassi della giornata
- A fine giornata il sistema deve elencare quali sono gli articoli da reintegrare in base alle vendite
- Il tempo di risposta a qualunque interrogazione deve essere inferiore a 3 secondi



# Spazio per migliorare

- Dominio
  - Un negozio contiene al massimo 20 registratori di cassa
  - Sintesi degli incassi = elenco dei totali incassati da ogni registratore di cassa a fine giornata
- Requisiti
  - Il sistema deve gestire tutti i registratori di cassa del negozio
  - Il sistema deve stampare la sintesi degli incassi della giornata
  - Il sistema mantiene un valore di scorta minima per ogni prodotto
  - A fine giornata il sistema deve elencare gli articoli sotto scorta minima

# Esercizio: Bancomat

Si consideri un sistema per il prelievo automatizzato di denaro contante (Bancomat). Per poter utilizzare il sistema l'utente deve possedere una tessera magnetica. Il sistema deve mettere a disposizione le funzioni di prelievo, saldo, estratto conto. Il sistema deve essere disponibile a persone portatori di Handicap, deve garantire un tempo di risposta inferiore al minuto, e deve essere sviluppato su architettura X86. Le operazioni di prelievo devono richiedere autenticazione tramite un codice segreto memorizzato sulla carta. Il sistema deve essere facilmente espandibile, e adattabile alle future esigenze bancarie.

## Convalida dei requisiti:

## Noam Chomsky, tecnica del metamodello in PNL

- I filtri linguistici sono i processi attraverso i quali costruiamo la nostra mappa del mondo:
  - Generalizzazione
  - Cancellazione
  - Deformazione

# Noam Chomsky: generalizzazione

- La **generalizzazione** è il processo attraverso cui le persone partendo da una esperienza specifica, la decontestualizzano traendone un significato universale.
- Per generalizzare un'esperienza utilizziamo:
  - quantificatori universali:
    - sempre, tutto, mai, nessuno, chiunque, ognuno, ...
  - operatori modali:
    - devo, posso, voglio.
- Per ricercare generalizzazioni
  - Sul serio tutti? davvero mai?

# Noam Chomsky: cancellazioni

- La **cancellazione** è un processo di selezione dell'esperienza. Le persone, infatti, prestano attenzione solo ad alcuni pezzi del proprio vissuto escludendone altri. Esempi:
  - cancellazione dell'indice referenziale: non è espresso l'attore dell'azione ("bisognerebbe agire")
  - comparativo mancante: quando un'affermazione è priva del termine di paragone ("voglio migliorare questa azienda")
- Per ricercare cancellazioni
  - Chi? Quando?

# Noam Chomsky: deformazioni

- La **deformazione** è una percezione "distorta" della realtà. Interpretiamo il mondo esterno secondo le nostre mappe, adeguando la realtà alle nostre attese. A volte utilizziamo un singolo episodio per distorcere totalmente una situazione che, invece, può essere interpretata e vissuta in modi diversi.
- Per ricercare cancellazioni
  - Davvero.... ?

# Documento dei requisiti in 5 passi

1. Acquisizione
2. Elaborazione
3. Convalida
4. Negoziazione
5. Gestione

# Negoziazione

- Modifiche alla lista di requisiti
- Divisione in classi (es. MoSCoW)
  - Must have (Requisiti obbligatori)
    - Irrinunciabili per il cliente
  - Should have (Requisiti desiderabili)
    - Non necessari, ma utili
  - Could have (Requisiti opzionali)
    - Relativamente utili, da realizzare se c'è tempo
  - Want to have (Requisiti postponibili)
    - Contrattabili per successive versioni



# Gestione dei requisiti:

## 1. identificazione

- Identificatore unico
  - chiave in un DB
- Numero sequenziale (1,2,3,4.....)
- Numero sequenziale basato sulla struttura del documento (2.4.7)
- Coppie <CATEGORIA, NUMERO>

# Gestione dei requisiti

## 2. attributi (esempio preso da RUP, Rational Unified Process, estensione di UP)

- Stato
  - Proposto, approvato, rifiutato, incorporato
- Vantaggi
  - Importanza, tipo MoSCoW
- Sforzo in gg/uomo
- Rischio
  - Valutazione della fattibilità tecnica
- Stabilità
- Versione destinazione
  - Per lo sviluppo incrementale

# Gestione dei requisiti:

## 3. tracciabilità

- Requisiti vs
  - componenti del sistema
  - moduli
  - test
- Strumenti CASE

# Requisiti e aspetti contrattuali

- Il documento dei requisiti normalmente precede la stipula del contratto, e ne è parte integrante
- Se alla stipula del contratto non è possibile avere un documento definitivo, è opportuno prevedere di rinegoziare il contratto.
- In caso di gara può essere prodotto dal committente come parte di capitolato tecnico.

# Riepilogo

- L'attività di analisi
- Descrizione del dominio
  - Modello del dominio
- Descrizione dei requisiti
  - Documento dei requisiti
- Requisiti funzionali e non funzionali

# Syllabus

- Dispense in copisteria

# Altri riferimenti

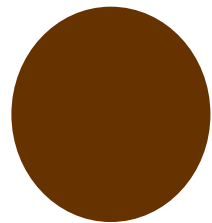
- L.A. Maciaszek, Requirements Analysis and System Design, Addison-Wesley
- E. Yourdon, Modern Structured Analysis, Prentice-Hall
- I. Sommerville, P. Sawyer, Requirements Engineering, Wiley
- G. Booch, OO Analysis and Design, Addison-Wesley

# Appendice: un esempio

- La missione GPL



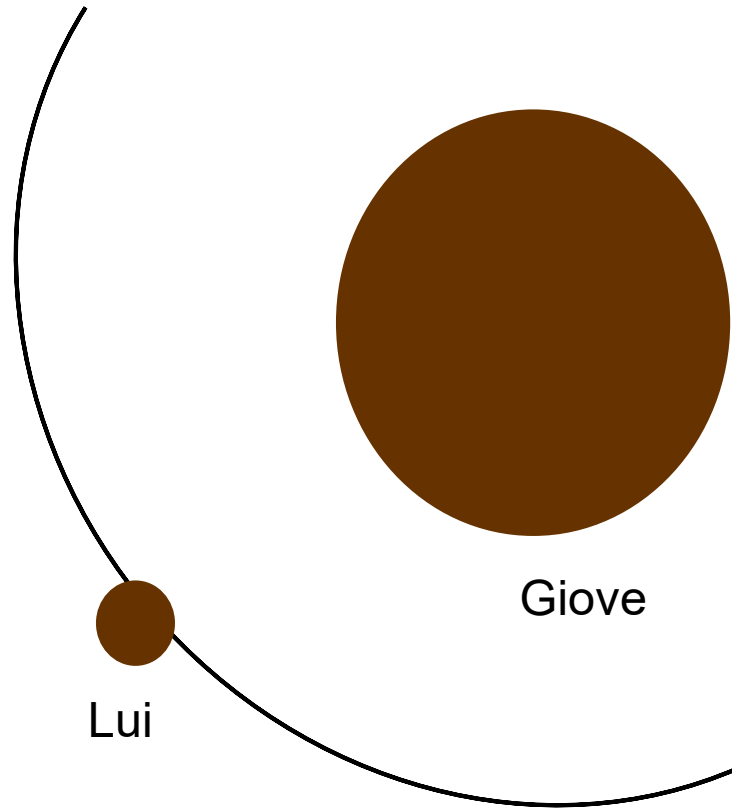
# La missione GPL: crociera



Terra



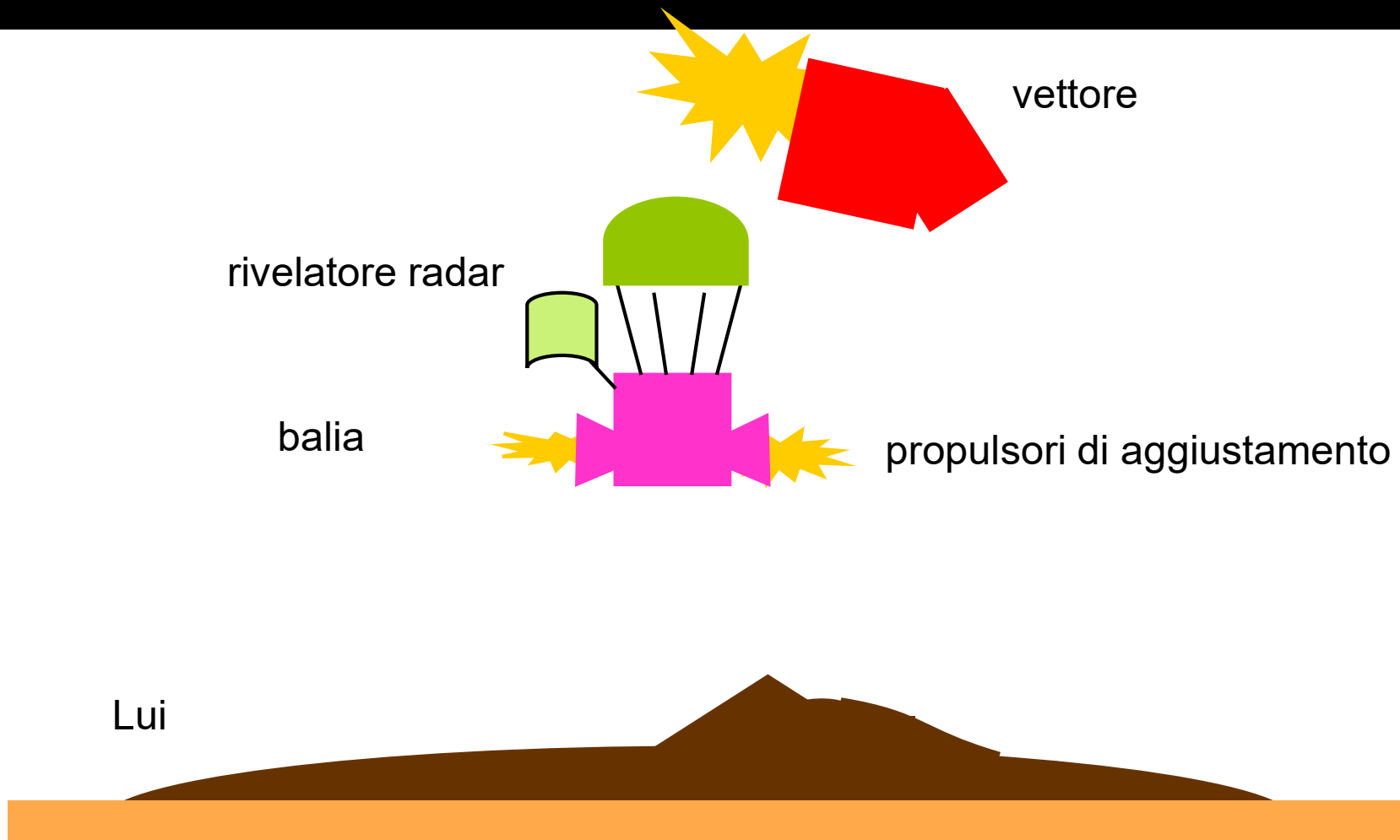
vettore



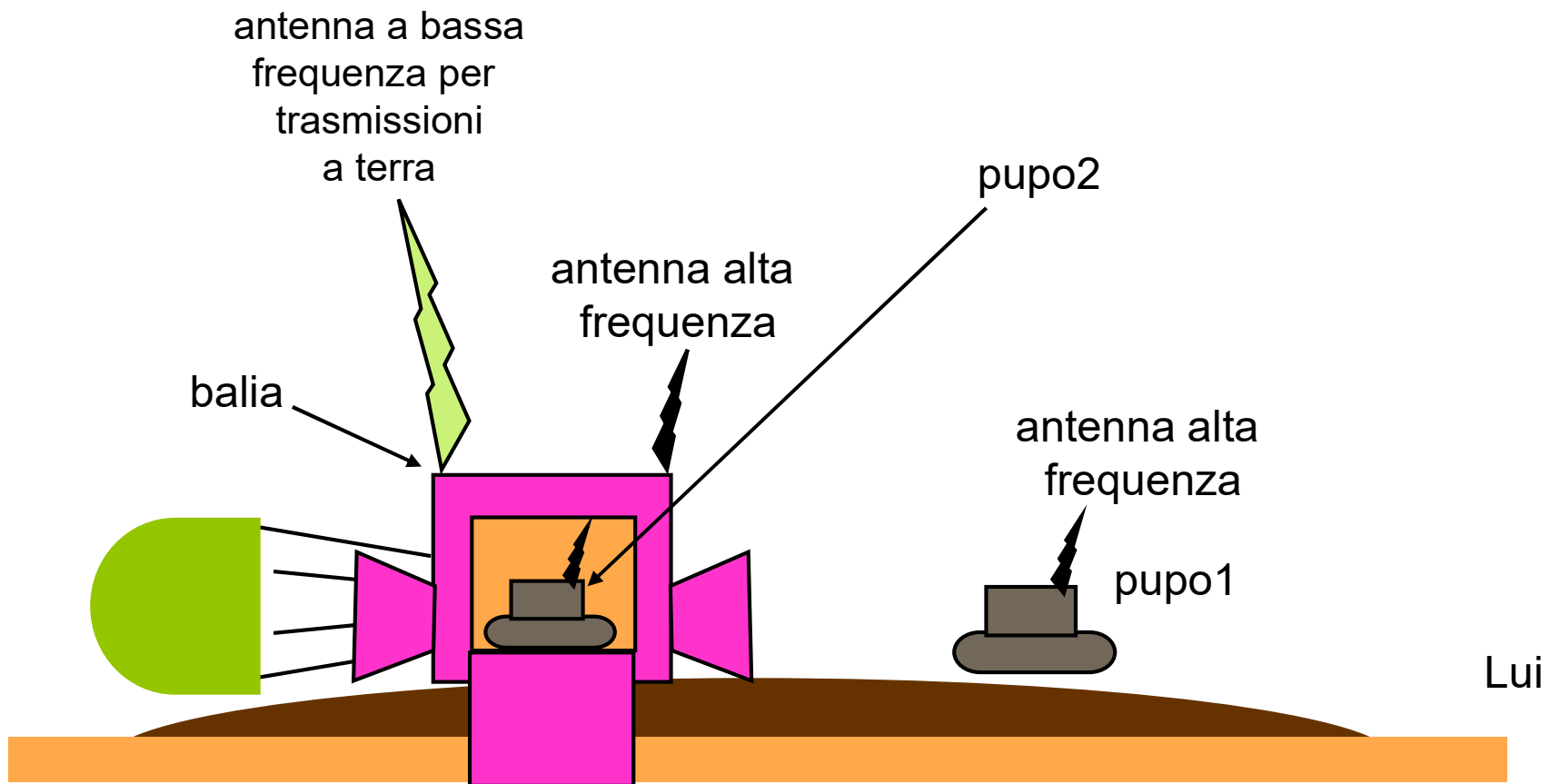
Lui

Giove

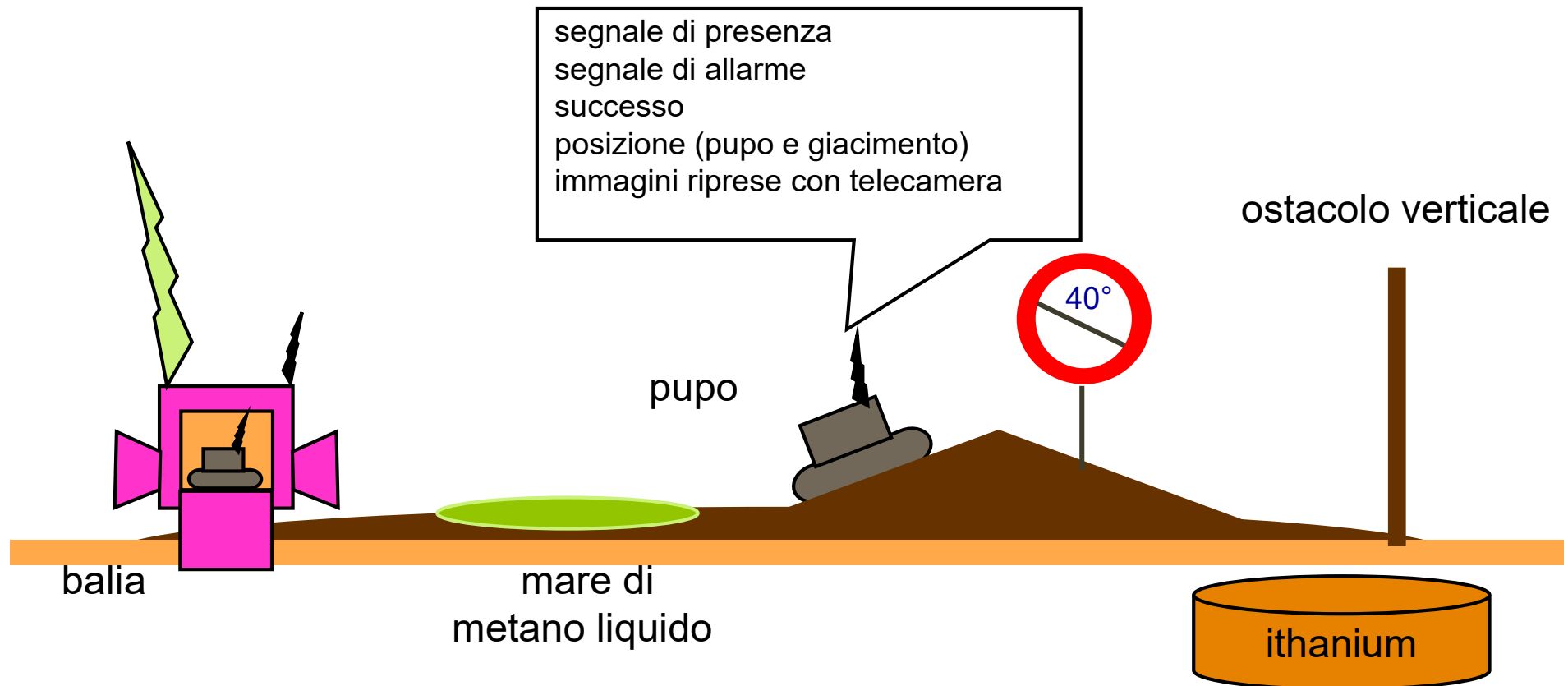
# La missione GPL: discesa



# La missione GPL: atterraggio



# La missione GPL: esplorazione



# Documentazione: testo originale

5. La missione GPL si compone di un modulo di atterraggio denominato BALIA (BASic Lander & Interface Adapter) e di due moduli di esplorazione semoventi denominati PUPO-1 e PUPO-2 (Prospection Unit POd 1 e POd 2); le due prospection unit sono identiche, e PUPO-2 viene mantenuto come backup nel caso PUPO-1 non sia più operativo.
6. La missione si svolge in due fasi: navigazione ed esplorazione.
7. *Fase di navigazione I.* Durante la crociera, PUPO-1 e PUPO-2 sono contenuti all'interno di BALIA, e nessuno dei moduli è operativo.  
[...]
14. Nel caso di atterraggio felice, BALIA apre il proprio vano carico e rende operativo PUPO-1.

# Costruzione glossario di dominio

- Si evidenziano i termini appartenenti al dominio
- 5. La missione GPL si compone di un modulo di atterraggio denominato BALIA (BASic Lander & Interface Adapter) e di due moduli di esplorazione semoventi denominati PUPO-1 e PUPO-2 (Prospection Unit POd 1 e POd 2); le due prospection unit sono identiche, e PUPO-2 viene mantenuto come backup nel caso PUPO-1 non sia più operativo.
- 6. La missione si svolge in due fasi: navigazione ed esplorazione.
- 7. *Fase di navigazione I.* Durante la crociera, PUPO-1 e PUPO-2 sono contenuti all'interno di BALIA, e nessuno dei moduli è operativo.
- 14. Nel caso di atterraggio felice, BALIA apre il proprio vano carico e rende operativo PUPO-1.

# Glossario di dominio

- Si riporta la definizione di ogni voce
- 23. **Missione.** È formata da una fase di **crociera** seguita da una fase di **esplorazione**. La fase di **crociera** è formata da una fase di **navigazione**, seguita da una fase di **discesa** e da un **atterraggio**.
- 9. **Balia.** Componente fisico. È formata da **paracadute**, **antenna a bassa frequenza**, **antenna ad alta frequenza**, **propulsori**, **portelli**, **vano carico**, **radar altimetrico**, **altimetro**, **orologio**, **hardware**. Al suo interno contiene **Pupo-1** e **Pupo-2**. Si trova su un punto della superficie caratterizzato da un'**altezza** e da un'**asperità**. Ha un'autonomia energetica sufficiente per tutta la durata della **missione**. Ha un hardware in grado di svolgere le funzioni richieste per la **missione**. Può atterrare su un punto della superficie in cui l'**asperità** è minore o uguale a 1.

# Estrazione dei requisiti: primo passo

- Si eliminano dal testo originale le parti relative al dominio
- 3. Dopo l'atterraggio, **Balia** apre il vano carico e rende operativo **Pupo-1**.
- 5. **Pupo** invia a **Balia** tre tipi di informazioni:
- 6. a. segnali di presenza, allarme e successo.
- 7. Il segnale di presenza (e unità operativa, detto anche heartbeat) deve essere inviato ogni secondo.
- 8. I segnali di allarme sono inviati quando il **Pupo** è in condizioni estreme, mentre il segnale di successo viene inviato quando il **Pupo** ha individuato un **giacimento di Ithanium** (devono essere inviate anche le sue coordinate);
- 9. b. dati di controllo, che includono la **posizione** corrente del **PUPO** relativamente alla **Balia**, e la condizione esterna in base ai dati dei **sensori**;
- 10. c. dati scientifici, che includono le **immagini**.



# Costruzione del glossario dei dati

## ■ Si evidenziano i dati

3. Dopo l'atterraggio, **Balia** apre il vano carico e rende operativo **Pupo-1**.
5. **Pupo** invia a **Balia** tre tipi di **informazioni**:
  6. **a. segnali di presenza, allarme e successo**.
  7. Il **segnale di presenza** (e unità operativa, detto anche **heartbeat**) deve essere inviato ogni secondo.
  8. I **segnali di allarme** sono inviati quando il **Pupo** è in condizioni estreme, mentre il **segnale di successo** viene inviato quando il **Pupo** ha individuato un **giacimento di Ithanium** (devono essere inviate anche le sue coordinate);
  9. **b. dati di controllo**, che includono la **posizione** corrente del **PUPO** relativamente alla **Balia**, e la condizione esterna in base ai dati dei **sensori**;
  10. **c. dati scientifici**, che includono le **immagini**.

# Glossario dei dati

- Si riporta la definizione di ogni voce
- 3. Dati di controllo. **Messaggio** di **classe b**. Include la **posizione** di **Pupo** e i dati rilevati dai suoi sensori. Invio ogni 10 secondi.
- 4. Dati scientifici. **Messaggio** di **classe c**. Include le **immagini**. Invio ogni 10 minuti.
- 6. **Messaggio**. Ha una **classe** di **priorità**. Ha un'indicazione dell'istante di creazione e primo invio
- 11. Segnale di presenza. **Messaggio** di **classe a**. Invio ogni secondo.

# Lista dei requisiti: software di Balia

1. Cercare un punto per l'atterraggio di Balia.
2. Dopo l'atterraggio aprire il vano carico e attivare Pupo-1.
3. Quando Pupo-1 è in avaria attivare Pupo-2.
4. Identificare le **condizioni rilevanti** di Balia.
5. Rispettare le **priorità** nell'invio a Terra dei **messaggi**.
6. Ritrasmettere a Terra il **segnale di allarme**.
7. Ritrasmettere a Terra il **segnale di successo**.
8. Ritrasmettere a Terra i **dati di controllo**.
9. Ritrasmettere a Terra i **dati scientifici**.
10. Inviare a Terra il **rapporto**.
11. Inviare o ritrasmettere a Terra i **segnali di avaria**.
12. Rilevare le **avarie dei sensori** e degli **attuatori** di Balia.
13. Svolgere le proprie funzioni con i **sensori** o gli **attuatori** di Balia in avaria.

# Lista dei requisiti: software di Pupo

1. Cercare uno o più giacimenti di Ithanium.
2. Raccogliere dati scientifici.
3. Acquisire i dati dei sensori di Pupo.
4. Identificare le condizioni estreme di Pupo.
5. Identificare le condizioni rilevanti di Pupo.
6. Inviare a Balia il segnale di presenza.
7. Inviare a Balia il segnale di allarme.
8. Inviare a Balia il segnale di successo.
9. Inviare a Balia i dati di controllo.
10. Inviare a Balia i dati scientifici.
11. Inviare a Balia i segnali di avaria.
12. Rilevare le avarie dei sensori e degli attuatori di Pupo.
13. Svolgere le proprie funzioni con i sensori o gli attuatori di Pupo in avaria.