

## Analisi e Specifica dei Requisiti (per riepilogare)

### Scopi della fase:

- Definizione di funzionalità, vincoli, prestazioni, interfacce e qualsiasi altra caratteristica che il sistema dovrà possedere per soddisfare le necessità del cliente
- Redazione di un Documento di Specifica dei Requisiti Software (SRS), che sia completo, preciso, consistente, non ambiguo, comprensibile sia al committente che allo sviluppatore
- Predisposizione di un piano di test e della versione 0 del manuale utente

*N.B. Definizione di COSA il sistema dovrà fare senza descrivere COME*

1

## L'importanza di un documento di SRS (per riepilogare)

- Un documento di Specifica dei Requisiti Software (SRS) costituisce il punto di convergenza di tre diversi punti di vista: cliente, utente, sviluppatore (negoziante)
- Un documento di SRS fornisce un punto di riferimento per la *convalida* del prodotto finale
- Un documento di SRS di qualità è il pre-requisito per un software di alta qualità
  - Un errore nell'SRS produrrà errori nel sistema finale
- Un documento di SRS di qualità riduce i costi di sviluppo
  - La rimozione di un difetto scoperto dopo lo sviluppo e rilascio del sistema può costare fino a 100 volte la rimozione di un difetto durante la fase di analisi e specifica dei requisiti

2

## Approcci di Analisi

### Approccio Informale

- nessun modello del sistema viene costruito (uso di un linguaggio informale)
- la raccolta delle informazioni avviene attraverso una serie di riunioni tra gli utenti/committenti ed analisti, l'uso di questionari, studio di documentazione esistente, ...
- si procede attraverso la formulazione e il raffinamento di vari documenti di SRS che sono sottoposti alla convalida nell'ambito di opportuni meeting

### Approccio basato su modelli concettuali

- produce rappresentazioni del dominio applicativo e del sistema basate su linguaggi formali o semi-formali inseriti nel documento

### Approccio basato sulla prototipazione

- il problema viene analizzato ed i requisiti sono compresi grazie all'uso di un prototipo del sistema da parte di cliente ed utenti

3

## Alcuni approcci basati su modelli concettuali

### Analisi Strutturata

- basata sull'uso di Data Flow Diagrams e Dizionario dei Dati
- l'analisi del problema viene eseguita usando l'approccio della decomposizione delle funzioni
- i dati e le relative relazioni sono modellati con linguaggi diversi (es. modello Entità-Relazioni)

### Analisi Object-Oriented

- l'analisi del problema viene eseguita usando l'approccio della decomposizione in oggetti (entità/ concetti del dominio del problema)
- uso di linguaggi di modellazione, es. UML

4

## Tecniche di Specifica

Tecniche diverse si adattano a diverse categorie di problemi

- alcune tecniche sono più potenti del necessario
- Per es. tecniche di specifica per sistemi real-time rispetto a sistemi sequenziali o concorrenti

È importante identificare la tipologia del problema per scegliere la tecnica più adatta

Le tecniche si suddividono

- rispetto al grado di **formalità** della descrizione (del rispettivo linguaggio)
- rispetto a cosa descrivono del sistema (il loro **stile**)

5

## Linguaggi di Specifica

Notazione informale: tipicamente linguaggio naturale

- linguaggio naturale spesso usato per esprimere i requisiti utente perché è accessibile al committente

Notazione semi-formale (ha sintassi ma poca semantica)

- immediatamente comprensibile ma ambigua
- notazioni grafiche accompagnate da descrizioni in linguaggio naturale
- i modelli semiformali sono allo stato dell'arte i più flessibili ed i più usati

Notazione formale (ha sia sintassi sia semantica rigorosa)

- elimina ambiguità
- aumenta capacità di astrazione
- consente di ragionare su e verificare proprietà, anche con **strumenti** (semi) automatici che supportano quel linguaggio

N. B. non tutte le proprietà sono formalmente verificabili

- Il progettista è il responsabile finale dell'adeguatezza della specifica

6

## Stili di Specifica

**Specifica operativa:** tratta il **comportamento** desiderato per il sistema

- Diagrammi del Flusso Dati (**DFD**),
- Macchine a Stati Finiti (**FSM**),
- Reti di Petri (**PN**)

**Specifica descrittiva:** tratta le **proprietà** desiderate del sistema

- Diagrammi Entità- Relazione (**ER**)
- Specifiche basate sulla Logica (**Trio**)

**Specifica mista:** soprattutto se in linguaggio naturale, è facile mescolare aspetti descrittivi e operazionali

- In **Z** fornisco un modello (una macchina astratta) poi descivo, facendo uso di logica, il meccanismo con cui funziona questa macchina astratta.

7

## Specifiche operazionali

La specifica fornisce una macchina astratta, un modello che si comporta come il sistema da costruire.

Suggerisce in sé un metodo per la realizzazione

Non è detto che indichi la soluzione migliore, ma sicuramente ne indica una corretta

**Pro**

- È facilmente eseguibile
- È facile costruire un prototipo astratto e verificarne l'aderenza alla specifica

**Contro**

- Spinge decisamente verso una certa implementazione
- È difficile risalire alle proprietà del sistema

8



## Specifica dei Requisiti

La fase di analisi non definisce tutti gli aspetti del software quali:

- interfacce utente, gestione dei casi di errore, vincoli di prestazioni, di progettazione, aderenza agli standard, ...

Pertanto:

La conoscenza acquisita nella fase di analisi dovrà essere tradotta nella specifica dei requisiti

- la traduzione non è immediata
- È necessario che l'SRS abbia un insieme di caratteristiche di qualità

13

## La Convalida delle Specifiche

➤ L'obiettivo è quello di assicurare che il documento di SRS rifletta accuratamente e con chiarezza i requisiti effettivamente richiesti al software

- Tipi di errori riscontrabili in un documento di SRS
  - *Omissione* (mancata presenza di un requisito)
  - *Inconsistenza* (contraddizione fra i vari requisiti o dei requisiti rispetto all'ambiente operativo)
  - *Incorrettezza* (fatti non corretti nell'SRS)
  - *Ambiguità* (requisiti con significati multipli)

*Possibile osservare il comportamento dinamico del sistema se si adotta un processo con prototipazione*

14

## Tecniche di convalida di un documento di SRS

- Revisioni ed ispezioni manuali
  - le tecniche più efficienti per la individuazione di errori/difetti
  - partecipano alle riunioni di revisione l'autore del documento, il cliente, un progettista, un esperto di qualità
  - ogni partecipante rivede il documento prima della riunione
  - uso di checklist durante l'ispezione
- Tecnica di Reading
  - qualcuno diverso dall'autore legge il documento per coglierne potenziali problemi (ambiguità, ...)
- Uso di scenari relativi alle varie modalità operative del sistema

15

## Alcune metodologie

- Data Flow Diagram (DFD)
- Object Oriented Modeling (OO)
- Use cases

Ci sono altre metodologie, come il Data Structured Systems Development (DSSD), Entity-Relationship (ER), il Structured Analysis and Design Technique (SADT)





16

## DFD

- I Data Flow Diagrams (DFD) possono usati per l'analisi e la definizione delle specifiche. Essi consentono di descrivere il flusso logico dei dati, senza entrare nel dettaglio delle procedure di elaborazione
- Un DFD non tenta di rappresentare le elaborazioni condizionali o i cicli. Semplicemente mostra il flusso dei dati.
- Un DFD vede un sistema come una funzione che trasforma l'input nell'output desiderato
- Un dato fornito in input, è sottoposto ad un insieme di trasformazioni prima di essere trasformato in output
- Un DFD cattura e rappresenta la trasformazione dei dati
- I componenti di un DFD modellano:
  - dati, processi, source di input, sink di output e frecce per rappresentare il flusso

17

## DFD Notation

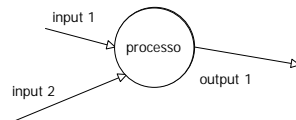
Process		Un transformer, una attività
Sink o source		Un agente esterno
File esterno		A data store
Operator And	*	 Un flusso di dati la freccia rappresenta la direzione del flusso A rappresenta un dato o una collezione di dati
Operator Or	+	

Da notare che ciascuna freccia deve avere una label!!!!

18

## Processo (Process)

- Rappresenta un'attività di trasformazione, che acquisisce dati in input e li trasforma in dati di output.
- Ogni processo:
  - deve essere collegato ad almeno un flusso di dati in input e ad almeno uno in output;
  - i flussi in output devono essere diversi rispetto ai flussi di input (in quanto oggetto di una trasformazione).



19

## Flusso di Dati (Data Flow)

- Indica un flusso di materiale (dati) omogeneo:
  - ha una direzione;
  - connette due elementi del sistema;
  - uno dei due elementi è necessariamente un processo (o un agente esterno) che produce il flusso o lo acquisisce.



20

## Deposito (Data Store)

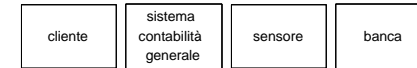
- È un archivio di dati permanenti, a cui i processi del sistema possono accedere, in lettura e/o in aggiornamento.
- È, per definizione, statico: mentre il flusso trasporta i dati, che sono quindi "in movimento", nel deposito i dati sono messi "a riposo", disponibili per essere trattati dai processi.



21

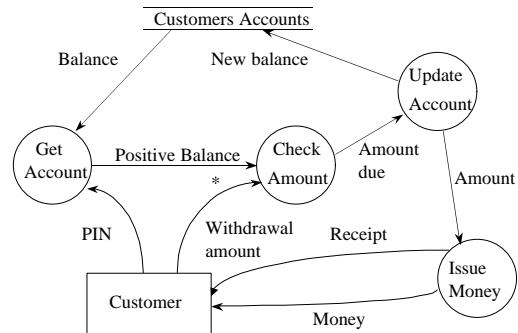
## Agente Esterno (External Agent)

- È un elemento/sistema esterno, con il quale il sistema da analizzare scambia informazioni in input e/o in output (sorgente o pozzo di flussi)
- Può essere una persona, un'organizzazione, un sistema hardware o software
- Come ogni sistema, potrebbe essere analizzato, ma:
  - l'agente esterno è da considerarsi come una "scatola nera", della quale non interessano le caratteristiche interne;
  - interessano solo gli scambi di dati (flussi) tra l'agente esterno ed il sistema da analizzare.



22

## Teller Machine



23

## Tracciare un DFD

- Identificare i principali input, i principali output e tracciare il *diagramma di contesto*:
  - esso rappresenta le interazioni tra il sistema e il "mondo esterno", in particolare esso prevede:
    - » un solo processo, che rappresenta il sistema nella sua globalità;
    - » tutti gli agenti esterni;
    - » i flussi che agenti esterni e sistema si scambiano;
    - » eventuali depositi.
  - esso descrive il sistema nella suo complesso; a partire da esso la metodologia basata sui DFD prevede che venga attuato un raffinamento successivo

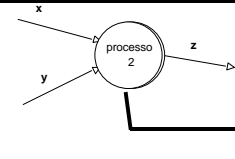
24

### Tracciare un DFD

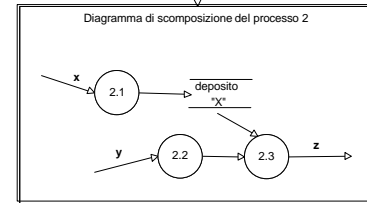
- Ogni processo può essere scomposto in sottoprocessi ('esplosione' di un processo), valgono le seguenti regole:
- la scomposizione origina un nuovo diagramma;
  - i flussi di input e di output collegati al processo "padre" devono essere collegati anche ai processi "figli" (padri e figli devono avere i medesimi input ed output "netti" - regola di continuità dei flussi);
  - la scomposizione è reversibile: è cioè possibile aggregare più processi in un macro-processo.

25

### Scomposizione dei Processi

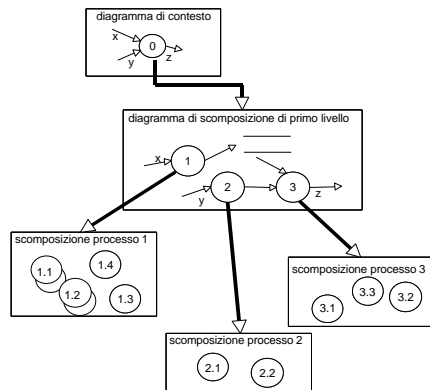


Ciascuna processo è identificato oltre che dal nome da un numero; i processi figli sono identificati con lo stesso numero del padre seguito da un altro numero progressivo, con una notazione 'punto'.



26

### DFD articolati su più livelli



27

### Dizionario dei Dati

- È un repository che deve accompagnare il DFD per definire tutti i flussi che in esso appaiono.
- La notazione è simile a quella delle espressioni regolari

Costrutto	Notazione	Significato
Definizione	A=B	A è definito come B
Sequenza	A+B	A è concatenato con B
Selezione	[A B]	A oppure B
Ripetizione	{A} <sup>j</sup>	A ripetuto da j (min) a i volte (max)
Opzionalità	[A]	A oppure nulla

Ad esempio  
 numero telefonico = [numero locale | numero esterno | centralino]  
 centralino=0  
 cifra={0|1|2...|9}  
 numero esterno={cifra}<sup>10</sup><sub>5</sub>

28

## Esempio di Dizionario dei Dati

Balance = user name + user account + user id + current balance

Withdrawal amount = [50000|100000|150000|200000|250000]+user id

New balance = user name + user account + user id + {digit}<sup>10</sup><sub>0</sub>

PIN = digit + digit + digit + digit + digit

29

## Errori commessi nei DFD

- Data flow non labellati
- Data flow mancanti: informazioni richieste da un processo che non sono disponibili
- Data flow estranei: informazioni che non sono usate nel processo
- Consistenza non conservata durante il raffinamento
- Processi mancanti
- Inserimento di incornamzioni di controllo

30

## Documento dei Requisiti

- È un documento ufficiale di cosa è richiesto allo sviluppatore del sistema
- Dovrebbe includere sia la definizione che la specifica dei requisiti
- Non è un documento di progetto; quindi dovrebbe rappresentare CHE COSA il sistema fa, e non che COME lo fa

È importante fornire una descrizione dei requisiti

- aiuta lo sviluppatore a comprendere il dominio di applicazione e perchè i requisiti sono stati presentati in una certa forma
- è particolarmente importante quando i requisiti devono essere modificati. La disponibilità della descrizione riduce la possibilità che la modifica possa causare degli effetti indesiderati

31

## Tracciabilità dei Requisiti

- È una proprietà della specifica dei requisiti che mette in relazione requisiti collegati in un modo tale che essi siano direttamente connessi con l'origine
- Da notare che si deve essere capaci di tracciare i requisiti negli componenti di progetto, nel codice, nei casi di test, ...

32

### Tipi di tracciabilità ...

- **Tracciabilità dell'origine:** indica l'origine di ciascun requisito: lo collega alla persone ovvero al documento da cui esso é sorto (ad esempio, il committente, un report, standard di qualità, regolamenti internazionali)
- **Tracciabilità della descrizione:** permette di definire perchè i requisiti sono stati specificati
- **Tracciabilità delle dipendenze:** collega i requisiti con altri requisiti, che sono in qualche modo dipendenti da essi

33

### ... tipi di tracciabilità

- **Tracciabilità dei sottosistemi:** collega i requisiti con i sotto-sistemi in cui essi sono implementati; permette di definire *who is going to develop what*. I requisiti sono suddivisi per categorie in base ai sotto-sistemi che li governano
- **Tracciabilità del progetto:** collega i requisiti con componenti specifici del sistema che sono usati per implementarli
- **Tracciabilità dell'interfaccia:** collega i requisiti con le interfacce interne ed esterne del sistema che premettono di utilizzarli

34

### Tecniche di tracciabilità

- Assegnare un numero univoco a ciascun requisito  
<tipo-requisito><numero-requisito>
- Collegare i requisiti ai vari aspetti considerati usando il numero unico
- Tracciare una tabella di tracciabilità che mette in relazione I requisiti identificati con uno o più aspetti del sistema o del suo ambiente

		Aspetti del sistema o del suo ambiente						
		A1	A2	A3	A4	A5	...	An
Requisiti	R1		✓		✓			
	R2		✓	✓				
	...							
	Rm	✓			✓	✓		

35