T4 Contenuto di un analisi dei requisiti
Presentate un indice di un documento di analisi dei requisiti e descrivete in modo sintetico contenuto e ruolo di ogni capitolo.

INDICE

T5 Utilizzo degli scenari nella definizione dei requisiti di Manutenibilità
Spiegate il requisito di non manutenibilità e spiegate come il concetto di scenario possa essere utilizzato nella specifica della manutenibilità di un prodotto software.

MANUTENIBILITA': sforzo necessario per applicare le modifiche (aggiunte, aggiornamenti, cancellazioni, rifacimenti) richieste al progetto in esame.
Esempio di applicazione:
M1: Scenario di manutenzione: *modifica*, sostituzione di un sensore esistente
M2: Scenario di manutenzione: *aggiunta* di nuovi sensori degli stessi tipi di quelli installati
M3: Scenario di manutenzione: *aggiunta* di nuovi sensori *di tipo diverso* rispetto a quelli installati
M4: Scenario di manutenzione: *aggiunta* di una nuova rete
M5: *Manutenibilità generale* del software.

Uno **scenario** d'utilizzo è un'insieme di informazioni che descrivono dal punto di vista dell'utilizzatore un particolare utilizzo del sistema.
Nel caso della manutenibilità, tramite l'utilizzo degli scenari si vogliono catturare tutti i particolari aspetti del sistema in qualche modo legati ad un componente mantenibile, ovvero modificabile nel tempo. (es: utilizzo del sistema: rilevazione ed acquisizione dati da sensori → scenario di manutenibilità (1): modifica, sostituzione di sensori - scenario di manutenibilità (2): …)

T6 Processi di progettazione
Illustrate i processi di progettazione per decomposizione, composizione e aggiunta e discuterne l'applicabilità

DECOMPOSIZIONE: da un modello a grossi blocchi si ricavano altri blocchi di dimensioni più piccole, seguendo un approccio top-down.
Utilizzato nelle situazioni in cui si desidera aumentare la precisione e il livello di dettaglio nella strategia di progettazione del sistema generale, andando ad affrontare la composizione e la realizzazione di eventuali sottosistemi costituenti quest'ultimo.
Dai livelli superiori (top) dei grandi blocchi, a quelli elementari di dettaglio (down) dei singoli componenti.

COMPOSIZIONE: da un modello costituito da tanti blocchi si ricavano blocchi di dimensioni più grandi, seguendo un approccio bottom-up. (es: reingegnerizzazione di un prodotto basata sull'integrazione di una collezione di componenti già esistenti)
Utilizzato quando, al contrario di come succedeva con la strategia precedente, si vuole allargare il dettaglio affrontando una vista d'insieme per verificare il comportamento e il funzionamento globale e l'integrazione dei singoli componenti. Dai livelli inferiori di dettaglio (bottom) a quelli superiori e composti d'insieme (up).

AGGIUNTA: viene per prima cosa progettato il cuore del prodotto. Poi, per aggiunte successive, vengono incollate altre funzioni al cuore del programma facendo evolvere il sistema.
Utilizzato per proteggere il cuore di sistema, dunque per rafforzare la sicurezza e il controllo dello stesso ed escluderlo dall'attacco esterno o dall'intaccamento dovuto, ad esempio, a funzionalità aggiuntive poco corrette. Ha il vantaggio di poter aggiungere continuamente nuove attività e mansioni i tutta sicurezza e nell'ordine desiderato.

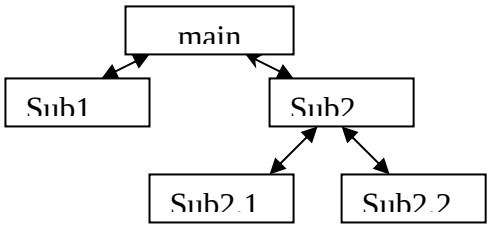
I processi non sono mutuamente esclusivi, dipendono dalla definizione dei requisiti e da altri fattori come tempo e costi. Si dovranno scegliere dunque il processo o i processi più confacenti al processo di progettazione.

T7 Stili di controllo
Presentate alcuni stili di controllo

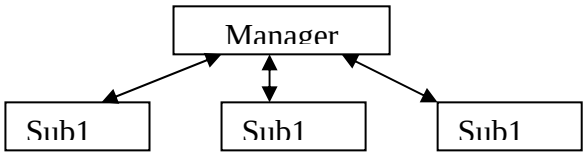
Schemi di gestione del flusso di controllo tra componenti.

- CONTROLLO CENTRALIZZATO

Call return
Gerarchia top-down di routines adatta a sitemi sequenziali



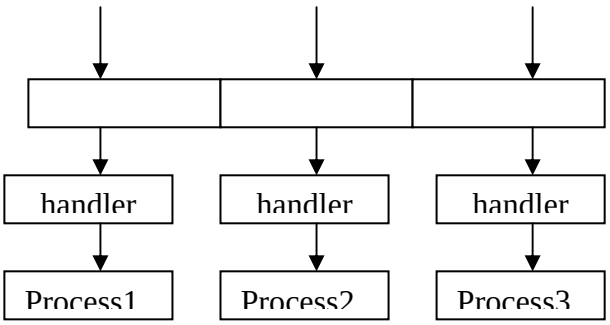
Manager
Un componente (manager) coordina l'esecuzione dei processi. Applicabile a sistemi concorrenti



- CONTROLLO BASATO SU EVENTI

Selective broadcast
. Ogni sottoinsieme registra l'**interesse** in uno specifico evento associando ad esso una procedura (1)
. Ogni sottosistema può annunciare un **evento** (2)
. Quando l'evento accade, il gestore **invoca** tutti i metodi (procedure) associati (3)
. Ogni sottosistema non sa se e quando un evento verrà gestito

Interrupt driven
. Ogni interrupt è associato ad un indirizzo e un handler
. Una commutazione attiva l'handler
. Applicato e utilizzato nei sistemi real-time



Interruptus vector

T8 Strategia di test
Un prodotto sw è costituito da 50 funzioni interattive che operano su un DB comune.
Alcune di esse sono combinatoriamente complesse e critiche per i danni economici che possono derivare da malfunzionamento.
1) presentate una bozza di possibile strategia di test di tipo funzionale (tecniche di test e criteri di copertura

Un sistema può essere costituito da più componenti interconnessi tra loro. E' possibile effettuare dei test anche a insiemi di componenti per verificarne il funzionamento collettivo. Si utilizzano metodi diversi: (TECNICHE DI TEST)

- **Big Bang Test**
  Il sistema viene integrato e poi verificato globalmente

- **Top-Down Testing**
  Si parte dal livello più alto del sistema, integrando i moduli uno alla volta, verificando passo per passo il loro funzionamento. E' possibile usare dei componenti fittizi (stub) che simulano il comportamento di moduli di basso livello non ancora implementati.
  Nella pratica questa tecnica può essere integrata con la bottom-up.

- **Bottom-Up Testing**
  Parte dal livello più basso del sistema, verificando i singoli componenti. Li si integra progressivamente analizzando i livelli man mano più alti. E' possibile usare componenti fittizi (drivers) che simulano il comportamento di moduli di alto livello non ancora integrati. Nella pratica questa tecnica può essere combinata con la Top-Down.

- **Driver e Stub**
  Componenti software sviluppati ad hoc per permettere il test di integrazione senza effettuare un'implementazione.

- **Test gerarchico**
  Si parte dallo strato più interno (per software con architettura a strati). Ogni strato deve essere testato dal punto di vista delle funzionalità attese dal suo utente, cioè lo strato superiore.

(CRITERI DI COPERTURA)

. Copertura del testing
Misura della quantità di controllo effettuata
. Copertura funzionale
Misura della quantità di funzionalità (definite dalla specifica) esercitate dal testing
n funzioni esercitate almeno una volta dal test / n tot di funzioni
. Copertura topologica
Misura della topologia e della quantità di strutture esercitate
n istruzioni esercitate almeno 1 volta dal test / n tot istruzioni

ESEMPIO DI STRATEGIA DI TEST

1. STRATEGIA DI TEST

E' stato adottato un **testing di tipo funzionale** la cui strategia è descritta schematicamente nei seguenti passi:

- Suddivisione del **piano di test** nelle aree funzionali del modello funzionale
- Estrazione delle funzionalità componenti ogni area funzionale
- Generazione di uno o più casi di test associati ad una funzionalità secondo seguenti criteri:

  - Definizione di un caso di test (positivo o negativo) per ogni **classe di equivalenza** dei dati in ingresso della funzionalità

  - Definizione di un caso di test (positivo o negativo) per ogni **condizione di confine tra classi di equivalenza** dei dati in ingresso della funzionalità

  - **Per specifiche funzionalità**: *esame delle condizioni (stati ed eventi) che possono influenzare l'esecuzione della funzione.*
    (ad esempio la contabilizzazione di un consumo può avvenire durante un periodo di fatturazione (stato) o durante il cambio del periodo di fatturazione (evento) che può coincidere con un cambio di contratto (eventi simultanei).
    Vengono considerate tutte le possibili combinazioni di condizioni ed i casi di test vengono organizzai per livelli successivi di approfondimento **)**

Al fine di identificare tutte le possibili combinazioni di eventi di interesse, è stato utilizzato lo strumento delle **"tavole di combinazione di eventi"** allegate al piano di test.

- All'insieme delle combinazioni di possibili condizioni è sovrapposta la presenza di un **power fail** e vengono generati i corrispondenti casi di test.

2. CRITERI DI COPERTURA DEL TESTING

Sono stati adottati i seguenti criteri di copertura minima dell'attività di test:

- Ogni funzione deve essere coperta da almeno un caso di test (copertura funzionale esaustiva)
- La distribuzione dell'intensità di test relativa alle aree funzionali (intensità di test = n. medio di test per ogni funzione dell'area funzionale) deve essere coerente con l'analisi di rischio effettuata.
- L'**analisi del rischio** identifica (e fornisce le ragioni della scelta) per ogni area funzionale un indice del rischio nella scala:
  1= rischio basso 2= rischio medio 3= rischio alto.

Ad ogni valore dell'indice di rischio viene associato un valore di intensità di test che deve essere soddisfatto (eventualmente in eccesso) dal piano di test, secondo il seguente criterio:

Livello di rischio 1: intensità di test 1
Livello di rischio 2: intensità di test 2
Livello di rischio 3: intensità di test 3

2) quale approccio utilizzereste per il test di accettazione presso il cliente?

TEST DI ACCETTAZIONE/COLLAUDO

La fase di rilascio di un prodotto software è una fase di specifica molto importante che deve essere gestita.
Dopo il test del prodotto in casa del produttore, possono essere eseguite più fasi diverse di verifica il cui termine è l'accettazione formale del prodotto da parte del cliente.

TEST	
?-TEST	all'interno del produttore che simula l'utente finale
?-TEST	presso il cliente in utilizzo controllato (es: n limitato di utenti)
AVVIAMENTO	preso il cliente in utilizzo normale, ma con sfiancamento del produttore
COLLAUDO	prove il cui esito positivo comporta l'accettazione del prodotto
ESERCIZIO	

T9 Test funzionali, strutturali e interazione

Spiegate a differenza tra test funzionali e test strutturali e possibili interazioni tra i due tipi di test.

TEST FUNZIONALE (BLACK BOX)

Provate le funzioni del programma indipendentemente da come il programma è costruito
Programma: scatola nera con ingressi e uscite.
Può essere implementato con le seguenti tecniche:
- casi d test da specifiche testuali
- casi di test da modelli
- classi di equivalenza del dominio di ingresso
- grafi causa-effetto

- casi di test da scenari

TEST STRUTTURALE (WHITE BOX)
Provate le strutture del programma (es: percorsi interni).
Si utilizza la conoscenza interna del programma (in particolare della sua composizione)
Viene realizzato con il supporto di grafi di controllo

- copertura delle istruzioni (statement test)
- copertura delle decisioni (branch test)
- copertura delle condizioni (condition test)
- copertura dei cammini (path test)
- copertura del grafo di cammini tra moduli

I due test non sono mutuamente esclusivi.
Sebbene il test funzionale è la tipologia maggiormente effettuata, il test strutturale viene effettuato la maggior parte delle volte per migliorare il test funzionale, infatti:
- una struttura dati è trattata da parti separate di codice in modi diversi
- si può misurare la copertura topologica ottenuta dal test funzionale e, se è il caso, intensificare il test funzionale.
- grazie al test strutturale è possibile rilevare i difetti alla base dei malfunzionamenti.

T10 WBS e stima di risorse in un progetto software
Discutere la relazione tra WBS d progetto e la stima delle risorse.

La stima delle risorse può essere realizzata in modalità diverse, tra cui, importante, la **stima analitica a partire da una WBS dettagliata**.

- Il progetto è scomposto in una serie di attività e componenti
- Si definiscono le caratteristiche dei componenti nel modo più analitico possibile
- Per ogni attività/componente è stimato lo sforzo necessario in termini di giorni/uomo e costo componenti/servizi acquisiti.
- La stima è effettuata con i migliori esperti disponibili
- Gli errori di stima, separando i lavori, tendono ad annullarsi reciprocamente (x componenti sovrastimati tendono a bilanciarsi con y componenti sottostimati)
- I programmatori tendono ad essere sempre ottimisti. Si deve valutare l'opportunità di inserire margini per gestire il rischio di difficoltà inattese.

Presentate e discutete alcune tecniche di stima delle risorse in un progetto software

- **Stima globale a partire da processi analoghi**
  Tecnica molto rischiosa e poco utilizzata

- **Tecnica DELPHI**
  La WBS viene effettuata da due persone diverse, in modo dettagliato e completo.
  I due si riuniscono e confrontano i risultati: se sono simili si valutano i dettagli, se invece sono significativamente diversi si discute fino a trovare un accordo sulle possibili soluzioni e sui motivi delle divergenze.

- **Modelli (COCOMO)**
  Formule parametriche che quantificano
  - M mesi d'uomo di sviluppo
  - T tempo in mesi di sviluppo

Valori dei parametri ricavati dalla classificazione ed analisi di progetti reali rappresentativi dei vari settori (avionica, EDP, …)
Richiede una base nota di progetti corrispondenti a quello da stimare per la taratura dei parametri.