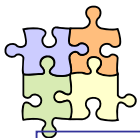


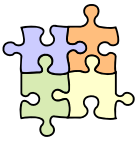
Requisiti e scenari

Dispensa AS 10
ottobre 2008



- Fonti

- ▣ [SSA] Chapter 8, Scope, Concerns, Principles, and Constraints
- ▣ [SSA] Chapter 10, Identifying and Using Scenarios
- ▣ [SAP/2e] Chapter 4, Understanding Quality Attributes
- ▣ [Ghezzi/2e] Capitolo 2, Il software: natura e qualità
- ▣ [Larman] Applicare UML e i pattern, Capitolo 6, Casi d'uso



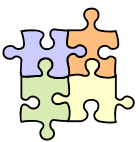
- Obiettivi e argomenti

□ Obiettivi

- comprendere i requisiti e come possono essere specificati
- comprendere gli scenari e la loro applicazione alla definizione dell'architettura

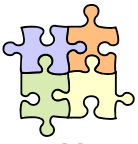
□ Argomenti

- requisiti
- scenari e architetture
- richiami: casi d'uso
- applicare gli scenari



* Requisiti

- Un'architettura deve soddisfare gli *interessi* e i *bisogni reali* delle *parti interessate*
 - quali sono le categorie di interessi e bisogni?
 - come possono essere descritti in modo efficace?
 - ad es., per la successiva progettazione e per consentire di valutare se un'architettura li soddisfa
 - come possono essere identificati e organizzati? come gestire la loro variazione ed evoluzione?
- Queste tematiche sono studiate, più in generale, nel mondo dell'ingegneria del software sotto le voci
 - *requisiti*
 - *ingegneria dei requisiti*
 - il processo di ricerca, analisi, organizzazione, documentazione, verifica e tracciatura dei requisiti (che cambiano)



Alcune possibili definizioni



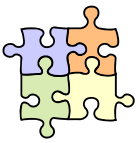
[Sommerville]

- i **requisiti** di un sistema sono la descrizione dei servizi forniti e dei suoi vincoli operativi



[Dorfman & Thayer]

- un **requisito** è una capacità del software richiesta dall'utente per risolvere un problema che consente di raggiungere un obiettivo
- una capacità del software che deve essere soddisfatta o posseduta da un sistema o componente di un sistema per soddisfare un contratto, uno standard, una specifica o quanto imposto da un altro documento formale

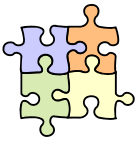


Gestione dei requisiti



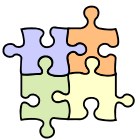
[Leffingwell & Widrig]

- la **gestione dei requisiti** è un approccio sistematico all'“elicitazione” (to elicit=far uscire, suscitare, dedurre, ricavare), organizzazione e documentazione dei requisiti del sistema, ed un processo che stabilisce e mantiene l'accordo tra il cliente e il team di progetto sui requisiti che cambiano del sistema



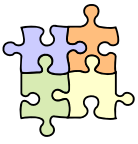
Alcune possibili classificazioni

- **Requisiti utente**
 - in linguaggio naturale – il contesto è la raccolta
- **Requisiti di sistema (specifici)**
 - definisce con precisione che cosa deve essere realizzato – il contesto è la definizione del contratto
- **Requisiti funzionali**
 - funzionalità che il sistema deve fornire
- **Requisiti non funzionali**
 - non riguardano le specifiche funzioni del sistema, ma piuttosto proprietà (qualità) del sistema nel suo complesso
- **Requisiti di dominio**
 - derivano dal dominio applicativo, e non da specifiche necessità degli utenti – possono essere sia funzionali che non funzionali



Requisiti non funzionali e qualità

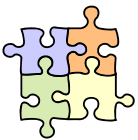
- I requisiti non funzionali sono anche chiamati **qualità**, **attributi di qualità** o **proprietà di qualità**
 - relativi ad interessi come sicurezza, prestazioni, usabilità, ...



Qualità

□ Qualità [Wikipedia]

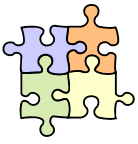
- Questa voce riguarda la qualità nell'accezione più usata, essenzialmente nell'ambito dell'ingegneria, dell'economia e della produzione, quando ci si riferisce ad un bene, materiale o immateriale, che viene prodotto per un determinato utilizzo.
- In generale, la misura della **qualità** indica una misura delle caratteristiche o delle proprietà di una entità (una persona, un prodotto, un processo, un progetto) in confronto a quanto ci si attende da tale entità, per un determinato impiego.
- L'uso che si intende fare è importante, poiché la valutazione della qualità varia a seconda dell'utilizzo. Per esempio, una persona può essere un ottimo scrittore, ma avere una valutazione molto bassa come atleta. Allo stesso modo, un gruppo di dati può avere un'alta qualità quando usati come informazione generica, divulgativa, ma una bassa qualità per un utilizzo di alta precisione.



Qualità

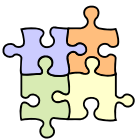
□ Diversi schemi di classificazione delle qualità

- qualità **interne** (ad es., modificabilità) ed **esterne** (ad es., prestazioni)
- qualità **del processo** (ad es., uso di un processo agile o burocratico) e qualità **del prodotto** (ad es., affidabilità)
 - le qualità del processo si riflettono in quelle del prodotto – ad es., un processo che prevede un'attività accurata di testing dovrebbe portare ad un'affidabilità migliore



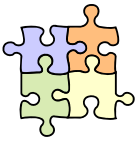
Principali qualità del software

- Principali qualità del software
 - correttezza – soddisfazione delle specifiche
 - una qualità assoluta (si/no)
 - non sempre ciò che si vuole – specifiche errate?
 - affidabilità – probabilità che non si verifichino malfunzionamenti
 - una qualità relativa
 - bisogna anche tenere conto della gravità dei malfunzionamenti che si possono verificare
 - robustezza – capacità del sistema di resistere a circostanze non desiderate
 - ad es., fallimenti di sottosistemi hardware o software
 - prestazioni e scalabilità
 - sicurezza
 - disponibilità



Altre qualità del software

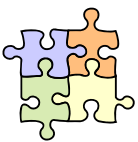
- Altre qualità del software
 - usabilità
 - costruibilità
 - verificabilità
 - manutenibilità
 - riparabilità – evolvibilità
 - riusabilità
 - portabilità
 - comprensibilità
 - interoperabilità
 - tempestività
 - visibilità (trasparenza del processo)
 - ...



Altre qualità del software

- Qualità di business
 - time to market
 - costi e benefici
 - vita attesa del sistema
 - programma dei rilasci
 - integrazione con sistemi legacy

- Altre qualità architettoniche
 - integrità concettuale

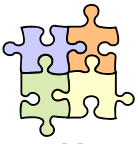


Funzionalità e qualità

- Funzionalità e qualità sono caratteristiche ortogonali
 - se ciò non fosse, la scelta delle funzioni avrebbe (ad es.) implicazioni sul livello di affidabilità o sulle prestazioni

- In generale (in teoria) è possibile scegliere indipendentemente le funzionalità, nonché un livello desiderato di ciascuna qualità
 - in pratica, non tutti i livelli di qualità sono raggiungibili con tutte le funzioni – ad es., elaborazioni complesse di immagini e prestazioni
 - non tutti i livelli di qualità sono sempre raggiungibili – ad es., sistema con elevate prestazioni, facilmente modificabile, poco costoso e prodotto in poco tempo

- L'architettura determina il livello relativo di ciascuna qualità



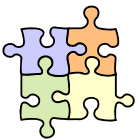
Interessi e requisiti

 [SSA]

- un **interesse** su un'architettura è un requisito, un obiettivo, un'intenzione o un'aspirazione che una parte interessata ha per quell'architettura
- Si tratta di una caratterizzazione piuttosto ampia

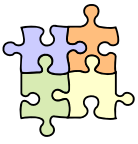
 [SSA]

- un **requisito** è un interesse specifico, non ambiguo e misurabile



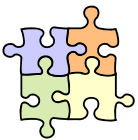
Interessi

- Può essere utile esprimere gli interessi a vari livelli di dettaglio/astrazione – esprimendo anche le correlazioni tra tali livelli – ad es., con una struttura ad albero
 - un **business goal** è un obiettivo specifico di un'organizzazione
 - un **business driver** è una forza che agisce sull'organizzazione, e richiede che essa si comporti in un certo modo
 - un **interesse** è un requisito, un obiettivo, un'intenzione o un'aspirazione che una parte interessata ha su un sistema/architettura



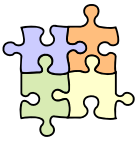
Interessi

- Si consideri un'organizzazione che attualmente vende i suoi prodotti in modo tradizionale
 - vuole iniziare a vendere i suoi prodotti mediante un servizio di commercio elettronico
 - business driver
 - bisogno competitivo dovuto al fatto che i principali concorrenti offrono, con successo, un servizio di questo tipo
 - business goal
 - nell'arco di un anno, il 15% delle vendite dovranno essere effettuate online
 - interessi
 - sito di commercio elettronico facilmente fruibile
 - l'esperienza di acquisto deve in qualche modo ricordare quella del negozio, per suggerire che si tratta di un modo alternativo per acquistare dalle stesse persone



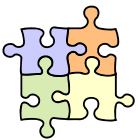
Dagli interessi ai requisiti

- I requisiti sono interessi specifici, non ambigui e misurabili
 - un esempio di interesse
 - sito di commercio elettronico facilmente fruibile
 - una descrizione un po' più precisa – ma ancora troppo vaga
 - un sito di commercio elettronico deve essere facile da usare da clienti che hanno esperienza limitata con i calcolatori e con il commercio elettronico
 - una descrizione ancora più dettagliata – ma non misurabile/verificabile
 - il processo di registrazione sul sito deve essere veloce e richiedere un data entry minimo
 - una descrizione ancora più dettagliata e verificabile
 - il processo di registrazione sul sito deve richiedere meno passi e l'immissione di un numero minore di dati rispetto ai siti concorrenti



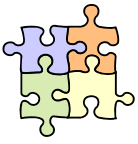
Requisiti e architetture

- Come già detto, l'architetto
 - non deve fare raccolta dei requisiti
 - è interessato ad una visione ampia
 - obiettivi, driver e interessi, delle varie parti interessate
 - interessi funzionali e di qualità
 - può essere interessato anche ad un insieme di requisiti – ma principalmente come esemplificazione degli interessi



* Scenari e architetture

- L'*applicazione degli scenari* alle architetture è una delle tecniche più potenti per sviluppare e validare un'architettura
 - scenari funzionali e scenari di qualità
 - per progettare/comprendere/descrivere/verificare come l'architettura funziona in pratica
- Prima di andare avanti, ricordiamo come vengono lavorati i casi d'uso in UP

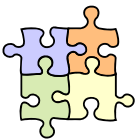


* Richiami: Casi d'uso

- I requisiti funzionali possono essere espressi sotto forma di casi d'uso
 - i **casi d'uso** sono storie scritte che raccontano come gli utenti (attori) possono raggiungere i loro obiettivi usando il sistema

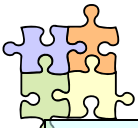
Elabora Vendita:

Un cliente arriva alla cassa con alcuni articoli da acquistare. Il cassiere usa il sistema POS per registrare ogni articolo acquistato. Il sistema mostra il totale e i dettagli per ogni articolo. Il cliente inserisce informazioni sul pagamento, che il sistema convalida e registra. Il sistema aggiorna l'inventario. Il cliente riceve dal sistema una ricevuta e poi se ne va con gli articoli acquistati.



Casi d'uso e scenari

- Un **attore** è qualcosa o qualcuno dotato di comportamento e che interagisce con il sistema
 - persone con un ruolo, sistemi informatici, organizzazioni
- Uno **scenario** (di caso d'uso) è una sequenza specifica di azioni e interazioni tra il sistema e alcuni attori
 - descrive una particolare storia nell'uso del sistema
 - uno scenario può essere di **successo** o di **fallimento**
- Un **caso d'uso** è un insieme di scenari correlati, di successo e fallimento, che descrivono un attore che usa un sistema per raggiungere un obiettivo (un risultato osservabile e di valore)



Esempio (in formato informale)

Gestisci Restituzione

Scenario principale di successo:

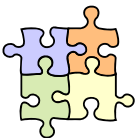
Un cliente arriva alla cassa con alcuni articoli da restituire. Il cassiere usa il sistema POS per registrare ciascun articolo restituito ...

Scenari alternativi:

Se il cliente aveva pagato con la carta di credito, e l'operazione di rimborso sulla carta di credito è stata respinta, allora il cassiere informa il cliente e lo rimborsa in contanti.

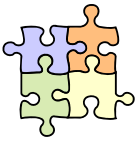
Se il codice identificativo dell'articolo non viene trovato nel sistema, il sistema avvisa il cassiere e suggerisce l'inserimento manuale del codice (può darsi che sia danneggiato).

Se il sistema rileva un fallimento nella comunicazione con il sistema esterno di gestione della contabilità, ...



- Casi d'uso e processo di sviluppo

- Un processo software (ovvero, alcune attività nello sviluppo del software) può essere guidato dai casi d'uso – ad es., UP
 - **requisiti** identificati e scritti come casi d'uso
 - quali le parti interessate? quali gli attori? quali gli obiettivi degli attori che useranno il sistema?
 - **progettazione orientata agli oggetti** guidata dai casi d'uso – realizzazioni di casi d'uso
 - che succede durante l'esecuzione di uno scenario di caso d'uso? quali oggetti sono coinvolti? come collaborano i vari oggetti?
 - l'obiettivo è progettare una decomposizione ad oggetti della logica applicativa
 - identificando gli oggetti, le loro responsabilità ed interfacce e le interazioni tra gli oggetti
 - è possibile ispirarsi al modello di dominio



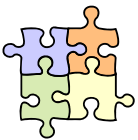
Casi d'uso e processo di sviluppo

- Anche la **progettazione dell'architettura** può essere guidata dai casi d'uso (e, più in generale, dagli scenari)
 - che succede durante l'esecuzione di uno scenario di caso d'uso? quali sottosistemi sono coinvolti? come collaborano i vari sottosistemi?
 - l'obiettivo è progettare una decomposizione in elementi dell'intero sistema
 - identificando gli elementi, le loro responsabilità ed interfacce e le interazioni tra gli elementi
 - domande a cui non sappiamo ancora rispondere
 - ma quale la natura degli elementi? quali le interazioni ammesse? quali criteri e linee guida? a che cosa è possibile ispirarsi nel progettare questa decomposizione?

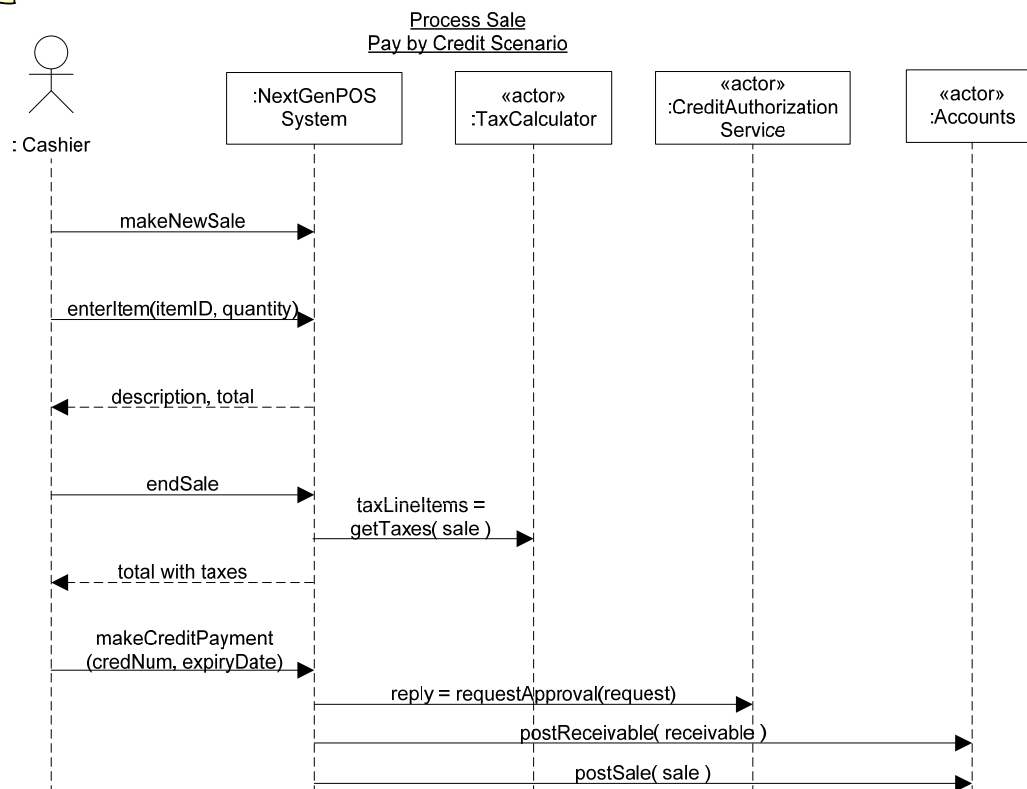
25

Requisiti e scenari

Luca Cabibbo - SwA



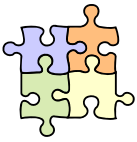
Esempio - collaborazione con sistemi esterni



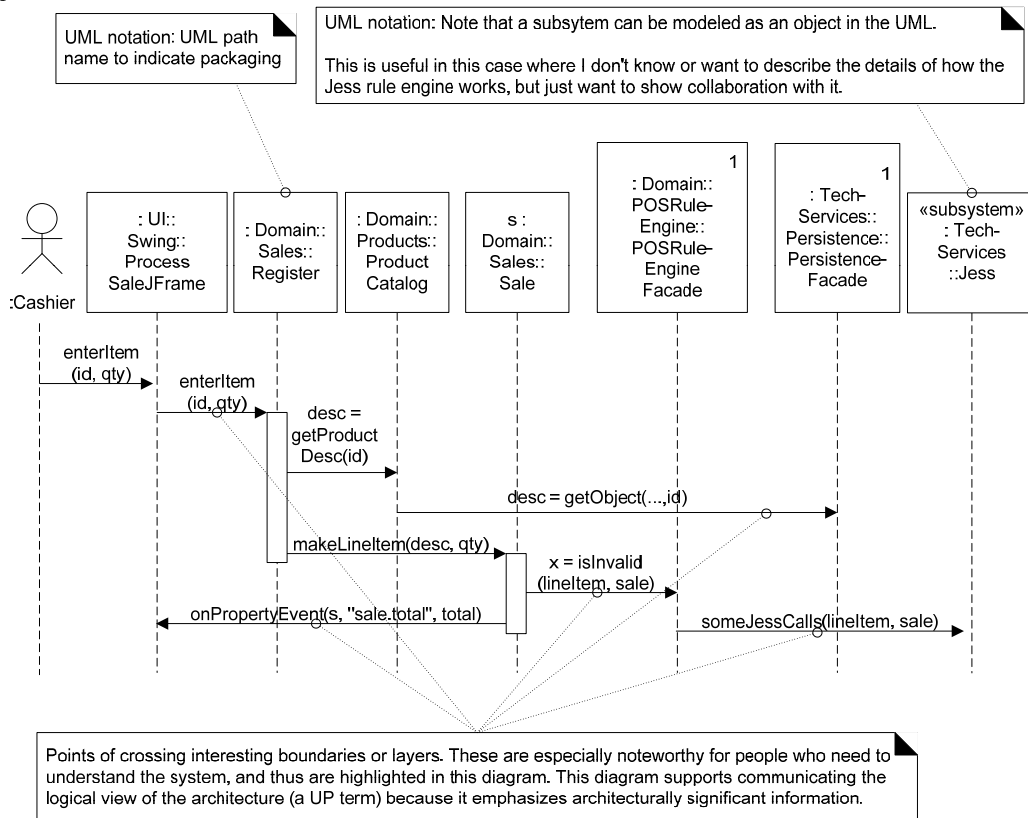
26

Requisiti e scenari

Luca Cabibbo - SwA



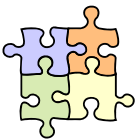
Esempio - collaborazione tra elementi interni



27

Requisiti e scenari

Luca Cabibbo - SwA



- Come trovare i casi d'uso

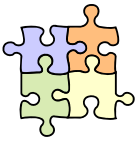
Strategia di base per l'identificazione dei casi d'uso

- determina i confini del sistema
 - l'applicazione software, l'intero sistema HW+SW, il sistema HW+SW e la persona che lo usa, l'intera organizzazione
 - questo passo è necessario perché i casi d'uso possono essere usati anche per descrivere sistemi non software
- identifica gli attori primari
 - coloro che usano il sistema – ciascuno per raggiungere i propri obiettivi
- identifica gli obiettivi di ciascun attore primario
- definisci i casi d'uso che soddisfano gli obiettivi degli utenti
 - dagli un nome con riferimento all'obiettivo
 - definisci uno scenario principale
 - identifica gli scenari alternativi

28

Requisiti e scenari

Luca Cabibbo - SwA



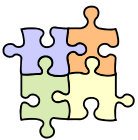
* Applicazione di scenari

- L'**applicazione degli scenari** alle architetture è una delle tecniche più potenti per sviluppare e validare un'architettura
 - scenari funzionali e scenari di qualità
 - per progettare/comprendere/descrivere/verificare come l'architettura funziona in pratica
 - generalizzazione della vista +1 del modello a 4+1 viste di Kruchten



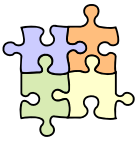
[SSA]

- uno **scenario architettuale** è una descrizione chiara e concisa di una situazione che il sistema dovrà probabilmente affrontare nel suo ambiente di produzione, insieme ad una definizione della risposta richiesta dal sistema



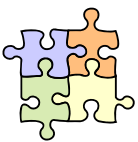
- Tipi di scenari

- Due gruppi di scenari
 - **scenari funzionali** – sequenze di eventi esterne a cui il sistema deve rispondere
 - ad es., scenari di casi d'uso
 - **scenari di qualità del sistema** – definiti in termini di come il sistema deve reagire a un cambiamento nel suo ambiente per esibire una o più proprietà di qualità
 - ad es., uno scenario di modificabilità – il sistema è in produzione – è richiesto che il sistema debba gestire un attributo in più (che può essere rappresentato come nuovo attributo di una relazione esistente) – il sistema deve poter essere modificato in due giorni lavorativi
 - raggruppati per (macro-obiettivo di) qualità
 - dettagliati in modo tale da essere verificabili



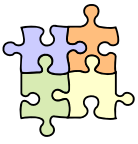
- Usi degli scenari

- Gli scenari possono essere usati in vari modi nel processo di definizione dell'architettura
 - per fornire un input alla definizione dell'architettura
 - per valutare l'architettura
 - per la comunicazione con le parti interessate
 - per trovare requisiti mancanti
 - per guidare il processo di verifica



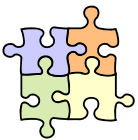
- Identificazione di scenari

- Gli scenari possono essere identificati a partire da
 - requisiti – l'identificazione dei requisiti può partire dall'identificazione delle parti interessate e proseguire con l'identificazione degli interessi di ciascuna parte interessata – espressi (se il caso) come scenari
 - parti interessate
 - esperienza – con progetti simili



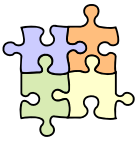
Prioritizzazione di scenari

- Nella definizione dell'architettura guidata da scenari
 - non tutti gli scenari sono ugualmente importanti
 - agli scenari va assegnata una *priorità* sulla base
 - del valore per le parti interessate
 - importanza, frequenza, valore economico, ...
 - del rischio
 - difficoltà tecniche, copertura dell'architettura, ...
 - bilanciando il grado di importanza relativa indicato dalle diverse parti interessate



- Descrizione di scenari

- Descrizione di *scenari funzionali*
 - sommario – breve descrizione di ciò che lo scenario dovrebbe illustrare
 - stato del sistema – prima dell'occorrenza dello scenario
 - ambiente del sistema
 - stimolo esterno – definizione di ciò che causa l'occorrenza dello scenario
 - risposta richiesta del sistema – spiegazione di come il sistema dovrebbe reagire allo scenario, dal punto di vista di un osservatore esterno
- Nella definizione dell'architettura, sono sufficienti descrizioni che riportano solo informazioni (architetturalmente) significative



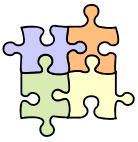
Esempio

- ❑ Scenario: Incremental Statistics Update
- ❑ Sommario
 - come il sistema gestisce un aggiornamento di una base di dati esistente
- ❑ Stato del sistema
 - esistono già dei dati aggregati per le vendite trimestrali a cui le statistiche incrementali si riferiscono
- ❑ Ambiente del sistema
 - il sistema è in esecuzione normale, nell'ambiente di produzione
- ❑ Stimolo esterno
 - arriva un aggiornamento relativo a parte delle transazioni di vendita del trimestre precedente, mediante l'interfaccia esterna *Bulk Load Data*
- ❑ Risposta richiesta del sistema
 - i dati entranti devono attivare automaticamente l'elaborazione delle statistiche in background, per aggiornare i dati aggregati per il trimestre interessato, per riflettere i nuovi dati – i dati aggregati precedenti devono rimanere disponibili fino a quando non sono pronti i nuovi dati aggregati



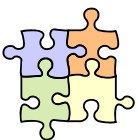
Descrizione di scenari di qualità

- ❑ Descrizione di *scenari di qualità del sistema*
 - sommario – breve descrizione di ciò che lo scenario dovrebbe illustrare
 - ambiente del sistema – con eventuale riferimento a fatto significativi circa lo stato iniziale del sistema
 - cambiamento dell'ambiente – spiegazione di ciò che è cambiato nell'ambiente e che causa l'occorrenza dello scenario
 - comportamento richiesto del sistema – definizione di come il sistema dovrebbe reagire al cambiamento dell'ambiente



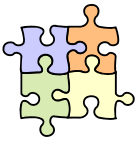
Esempio

- ❑ Scenario: Daily Data Update Trebles (triplo) in Size
- ❑ Sommario
 - come il sistema si comporta quando il volume regolare dei dati viene ecceduto in modo significativo
- ❑ Ambiente del sistema
 - il sistema è in esecuzione normale, nell'ambiente di produzione – esistono già dei dati aggregati nella base di dati – il volume giornaliero normale dei dati è di 1-1.5GB – in questo caso l'elaborazione richiede circa 3-4 ore
- ❑ Cambiamento dell'ambiente
 - arriva un aggiornamento giornaliero di 4GB
- ❑ Comportamento richiesto del sistema
 - il sistema deve elaborare i dati fino a quando il periodo di elaborazione non eccede un tempo prestabilito (e configurabile) – a quel punto l'elaborazione dei dati deve essere interrotta, devono essere mantenute le statistiche precedenti, e lasciato un messaggio diagnostico sulla console del sistema



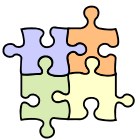
Esempio

- ❑ Scenario: Failure in Summary Database Instance
 - cambiamento dell'ambiente
 - nello scrivere le nuove statistiche nella base di dati, il sistema riceve un'eccezione (ad es., la base di dati è piena)
 - ...



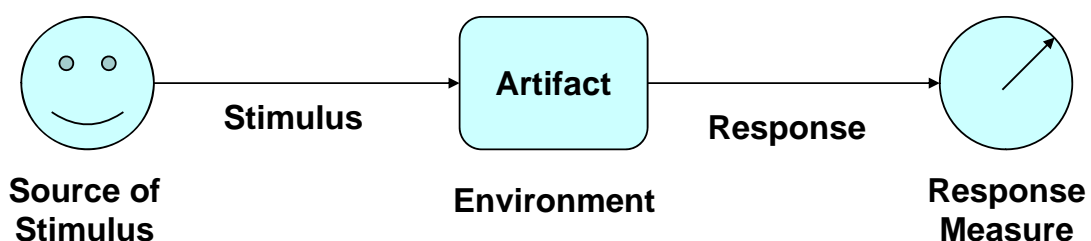
Esempio

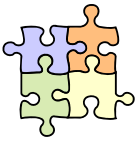
- Scenario: Additional Summary Dimension Required
 - cambiamento dell'ambiente
 - bisogna gestire una nuova dimensione rispetto alla quale aggregare i dati
 - comportamento richiesto del sistema
 - il team di sviluppo deve essere in grado di aggiungere la caratteristica richiesta con uno sforzo complessivo inferiore a un mese-uomo, senza cambiare la struttura complessiva del sistema



- Scenari per attributi di qualità

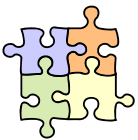
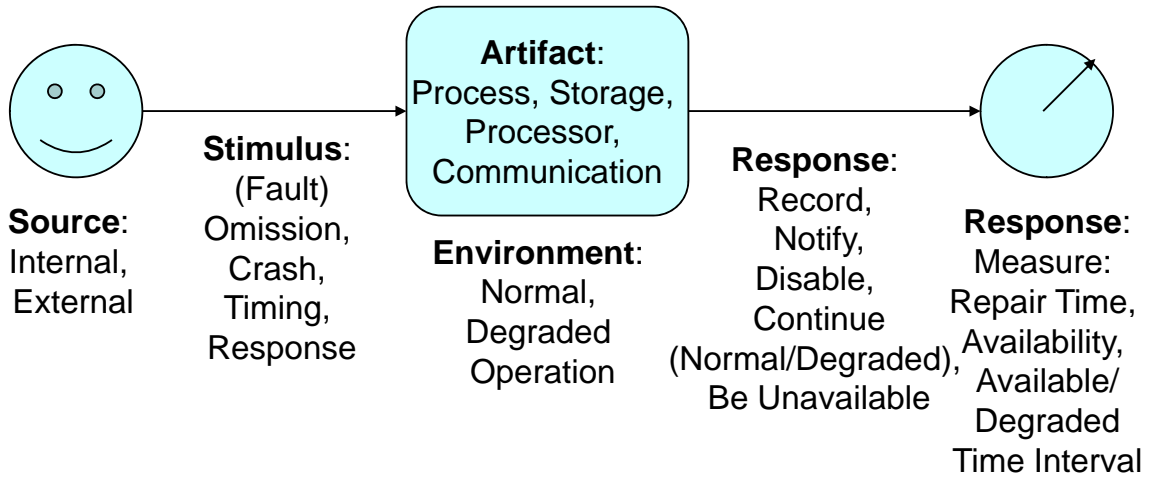
- [SAP/2e] descrive diversamente gli scenari di qualità
 - sorgente dello stimolo
 - stimolo
 - ambiente
 - elaborato – parte del sistema soggetta allo stimolo
 - risposta
 - misura della risposta



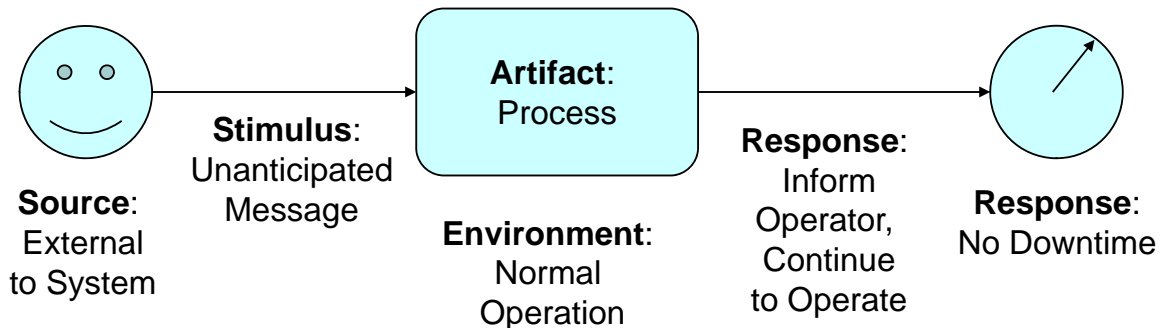


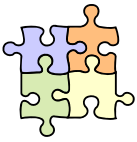
Scenari per attributi di qualità

- Inoltre, [SAP/2e] identifica alcune classi di qualità (ad es., disponibilità) e propone possibili valori per i vari parametri della descrizione
 - offre dunque un catalogo di scenari (parametrici) predefiniti



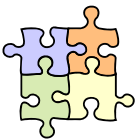
Esempio





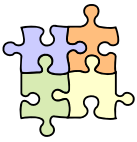
* Applicare gli scenari

- Ci sono vari modi per applicare gli scenari
 - ovvero, per descrivere come l'architettura reagisce/deve reagire allo stimolo che caratterizza lo scenario
- Attenzione, non tutti gli scenari sono importanti allo stesso modo
 - le tecniche più costose e precise vanno applicate solo nei casi veramente importanti, relativi ad aree di rischio maggiore

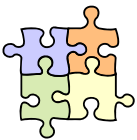
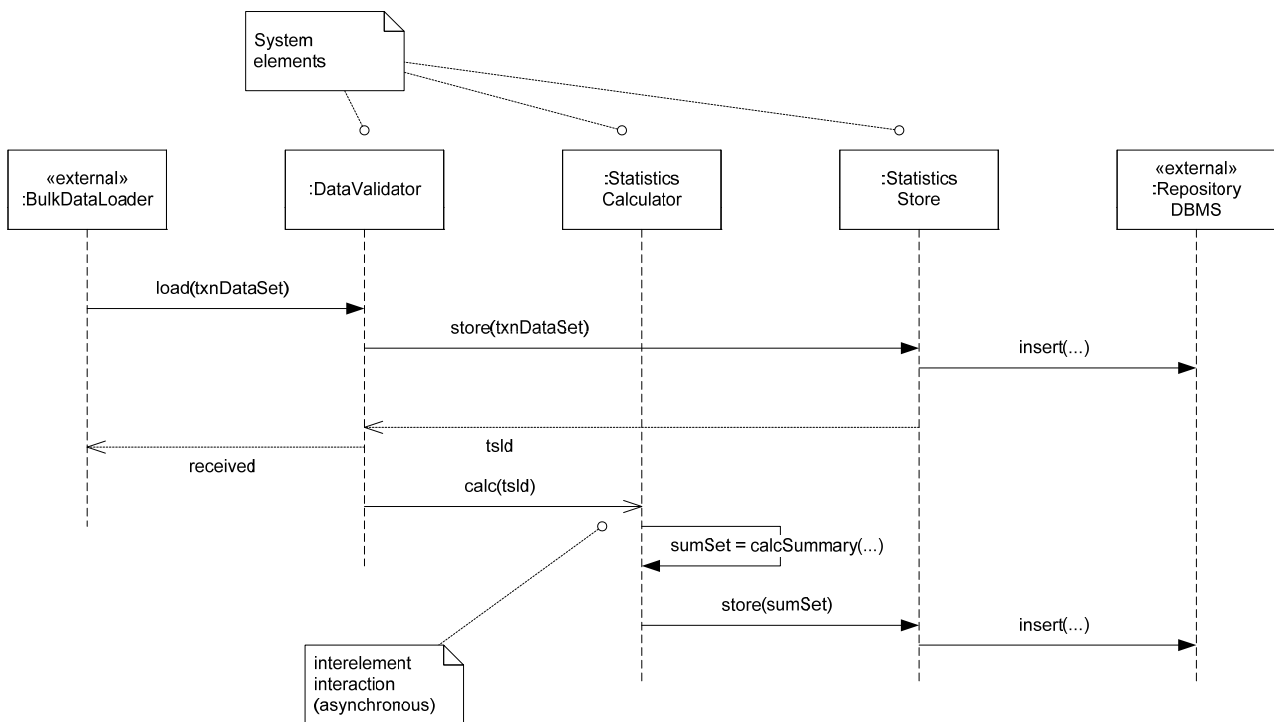


- Modelli

- Modelli (cartacei)
 - uno scenario viene usato durante la creazione di un modello
 - ad es., diagrammi di interazione di UML o descrizione dei flussi di dati/controllo
- Conseguenze
 - ☺ semplici da creare e comprendere
 - ☹ sono modelli inerti, che possono essere solo rivisti – ma non verificati

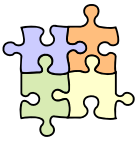


Esempio



- Walkthrough

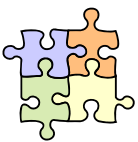
- I modelli “cartacei” possono essere valutati e validati rivedendoli in un gruppo di persone
 - **walkthrough** – significa “passeggiata” o “prova” (nel senso di prova teatrale)
 - una presentazione formale, con un oratorio competente, in cui viene illustrato come i vari elementi del sistema si comportano nella gestione di uno o più scenari
 - gli scenari guidano l’organizzazione della presentazione
 - i partecipanti possono fare domande ed obiezioni
 - simile ad una revisione formale di un documento – ma l’ordine della presentazione è guidato dagli scenari e non dalla sequenzialità del documento
 - una variante – gli elementi possono essere “impersonati”



Walkthrough

□ Conseguenze

- 😊 efficace nel trovare difetti e dimenticanze
- 😞 organizzazione complessa, poiché i partecipanti devono essere ben preparati
- 😞 i risultati effettivi dipendono dalla preparazione dei partecipanti



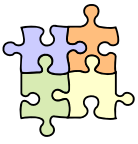
- Simulazioni

□ Simulazioni

- basate su opportuni modelli matematici
- ci sono strumenti di modellazione grafica (ad es., UML) che consentono alcune simulazioni
- ci sono simulatori dedicati ad alcune qualità – ad es., prestazioni

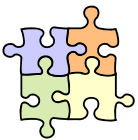
□ Conseguenze

- 😊 approccio efficace in alcuni casi, e meno costoso dello sviluppo di prototipi
- 😞 i risultati sono realistici? i risultati non possono essere riutati in fasi successive dello sviluppo (un prototipo potrebbe esserlo)



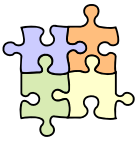
Altri modi di applicare gli scenari

- Prototipazione
- Conseguenze
 - 😊 può fornire un livello di confidenza elevato – da usare nei casi di rischio più elevato
 - ☹️ costoso



Altri modi di applicare gli scenari

- Test del sistema effettivo
 - gli scenari possono fornire la base per pianificare i test di sistema (di accettazione) del sistema reale



- Uso efficace di scenari

- Alcune buone pratiche per l'applicazione efficace degli scenari
 - identifica un insieme focalizzato di scenari
 - utile trovare molti degli scenari – ma è bene concentrarsi principalmente su quelli più importanti
 - usa scenari distinti
 - molti scenari possono variare tra loro di poco
 - utile identificare gruppi di scenari, con alcuni scenari principali – dando minore importanza alle variazioni
 - usa gli scenari presto
 - quando l'architettura sta prendendo forma – difficile imporre scenari significativi in un secondo momento
 - usa anche scenari di qualità – oltre a quelli funzionali
 - usa anche scenari di fallimento
 - coinvolgi le parti interessate