



UNIVERSITÀ DEGLI STUDI DI SALERNO

1

BASI DI DATI

Metodologie e Modelli per
progetti software



Interazione: processi software



Giuseppe D'Aniello
Researcher, PhD – IEEE Member

1

BASI DI DATI

Metodologie e modelli per i progetti software

2

DOMANDA

➤ Quali sono le attività principali comuni a tutti i processi software?

2

BASI DI DATI

3

Metodologie e modelli per i progetti software

RISPOSTA

➤ Quali sono le attività principali comuni a tutti i processi software?

- ✓ Analisi e specifica dei requisiti
- ✓ Progettazione e sviluppo del software
- ✓ Convalida del software
- ✓ Evoluzione del software

3

BASI DI DATI

4

Metodologie e modelli per i progetti software

ESERCIZIO

Esercizio 2.1 da "Ingegneria del Software, Ian Sommerville, 10ª edizione

➤ Motivando la vostra risposta in base al tipo di sistema che si sta sviluppando, suggerite i modelli generici di processo software più adatti per gestire lo sviluppo dei seguenti sistemi:

- 1) un sistema per controllare l'antibloccaggio dei freni in un'autovettura;

4

BASI DI DATI

5

Metodologie e modelli per i progetti software

SOLUZIONE

➤ **Motivando la vostra risposta in base al tipo di sistema che si sta sviluppando, suggerite i modelli generici di processo software più adatti per gestire lo sviluppo dei seguenti sistemi:**

1) un sistema per controllare l'antibloccaggio dei freni in un'autovettura

- È un sistema safety-critical che richiede una fase di analisi molto approfondita prima dell'implementazione.
- Ha bisogno sicuramente di un approccio allo sviluppo in cui i requisiti siano analizzati accuratamente
- Un modello a cascata è pertanto il più appropriato, magari anche con trasformazioni formali tra le diverse fasi di sviluppo

5

BASI DI DATI

6

Metodologie e modelli per i progetti software

ESERCIZIO

➤ **Motivando la vostra risposta in base al tipo di sistema che si sta sviluppando, suggerite i modelli generici di processo software più adatti per gestire lo sviluppo dei seguenti sistemi:**

2) un sistema di realtà virtuale per supportare la manutenzione del software

6

BASI DI DATI

7

Metodologie e modelli per i progetti software

SOLUZIONE

➤ **Motivando la vostra risposta in base al tipo di sistema che si sta sviluppando, suggerite i modelli generici di processo software più adatti per gestire lo sviluppo dei seguenti sistemi:**

2) un sistema di realtà virtuale per supportare la manutenzione del software

- È un sistema dove i requisiti cambieranno spesso e ci saranno componenti dell'interfaccia complesse
- Lo sviluppo incrementale, magari unito alla prototipizzazione rapida, è il più appropriato. (anche un processo Agile può essere adeguato)

7

BASI DI DATI

8

Metodologie e modelli per i progetti software

ESERCIZIO

➤ **Motivando la vostra risposta in base al tipo di sistema che si sta sviluppando, suggerite i modelli generici di processo software più adatti per gestire lo sviluppo dei seguenti sistemi:**

3) un sistema di contabilità universitaria che sostituisce il sistema esistente

8

BASI DI DATI

9

Metodologie e modelli per i progetti software

SOLUZIONE

➤ **Motivando la vostra risposta in base al tipo di sistema che si sta sviluppando, suggerite i modelli generici di processo software più adatti per gestire lo sviluppo dei seguenti sistemi:**

3) un sistema di contabilità universitaria che sostituisce il sistema esistente

- È un sistema i cui requisiti sono piuttosto noti e che sarà usato in un ambiente in cui coesisteranno molti altri sistemi come il sistema di gestione dei progetti di ricerca, gestione del personale, etc.
- Pertanto, un approccio basato sul riuso (e.g. riuso dei componenti) è quello più appropriato.

9

BASI DI DATI

10

Metodologie e modelli per i progetti software

ESERCIZIO

➤ **Motivando la vostra risposta in base al tipo di sistema che si sta sviluppando, suggerite i modelli generici di processo software più adatti per gestire lo sviluppo dei seguenti sistemi:**

4) un sistema interattivo di pianificazione dei viaggi che permette agli utenti di programmare i viaggi con il minore impatto ambientale

10

BASI DI DATI

11

Metodologie e modelli per i progetti software

SOLUZIONE

➤ **Motivando la vostra risposta in base al tipo di sistema che si sta sviluppando, suggerite i modelli generici di processo software più adatti per gestire lo sviluppo dei seguenti sistemi:**

4) un sistema interattivo di pianificazione dei viaggi che permette agli utenti di programmare i viaggi con il minore impatto ambientale

- Il sistema ha una interfaccia utente piuttosto complessa ma che deve essere stabile e affidabile.
- Pertanto, un approccio con lo sviluppo incrementale può essere appropriato poiché i requisiti cambieranno spesso quando l'utente comincerà a utilizzare il sistema realmente.

11

BASI DI DATI

12

Metodologie e modelli per i progetti software

DOMANDA

➤ **Perché nel modello a cascata il numero di iterazioni è limitato?**

12

BASI DI DATI

13

Metodologie e modelli per i progetti software

RISPOSTA

➤ Perché nel modello a cascata il numero di iterazioni è limitato?

Il modello a cascata è un modello document-driven in cui documenti (spesso formali e che necessitano di essere revisionati e approvati da più persone) sono realizzati alla fine di ogni fase.

A causa del costo di produrre e approvare i documenti, le iterazioni possono essere costose e aumentare eccessivamente i tempi del processo, anche senza mostrare avanzamenti visibili del processo. Per tale motivo, dopo un certo numero di iterazioni, si decide di continuare con il processo per poter arrivare al risultato finale.

Ciò può pregiudicare la qualità del software e la sua capacità di rispondere realmente ai requisiti dell'utente.

13

BASI DI DATI

14

Metodologie e modelli per i progetti software

DOMANDA

➤ Quali sono tre benefici dello sviluppo incrementale confrontato con il modello a cascata?

14

BASI DI DATI

15

Metodologie e modelli per i progetti software

RISPOSTA

➤ **Quali sono tre benefici dello sviluppo incrementale confrontato con il modello a cascata?**

- 1) Il costo per rispondere e recepire i cambiamenti dei requisiti utente è ridotto
- 2) È più semplice ricevere feedback del cliente sulle funzionalità già sviluppate
- 3) Delivery e deployment di funzionalità software al cliente più rapida

15

BASI DI DATI

16

Metodologie e modelli per i progetti software

ESERCIZIO

Esercizio 2.6 da "Ingegneria del Software, Ian Sommerville, 10ª edizione

➤ **Spiegate perché i cambiamenti sono inevitabili nei sistemi complessi e fornite esempi di attività di un processo software che possono aiutare a predire i cambiamenti e rendere il software che si sta sviluppando maggiormente resiliente ai cambiamenti.**

16

BASI DI DATI

17

Metodologie e modelli per i progetti software

SOLUZIONE

Esercizio 2.6 da "Ingegneria del Software, Ian Sommerville, 10° edizione

➤ **Spiegate perché i cambiamenti sono inevitabili nei sistemi complessi e fornite esempi di attività di un processo software che possono aiutare a predire i cambiamenti e rendere il software che si sta sviluppando maggiormente resiliente ai cambiamenti.**

- 1) I sistemi devono cambiare perché quando essi vengono installati in un ambiente, l'ambiente si adatta ad essi e questo adattamento genera naturalmente nuovi requisiti per il sistema.
- 2) Inoltre, l'ambiente è dinamico e cambia costantemente generando nuovi requisiti come conseguenza dei cambiamenti all'azienda, agli obiettivi aziendali, alle policy. A meno che il sistema non è adatto a riflettere questi requisiti, le sue funzionalità diventeranno obsolete rispetto alle funzionalità che servono per supportare il business e quindi diventa meno utile.

17

BASI DI DATI

18

Metodologie e modelli per i progetti software

SOLUZIONE

Esercizio 2.6 da "Ingegneria del Software, Ian Sommerville, 10° edizione

➤ **Spiegate perché i cambiamenti sono inevitabili nei sistemi complessi e fornite esempi di attività di un processo software che possono aiutare a predire i cambiamenti e rendere il software che si sta sviluppando maggiormente resiliente ai cambiamenti.**

- 3) Esempi di attività di un processo software che può supportare i cambiamenti:
 - Registrare le motivazioni dei requisiti cosicché la ragione per cui un requisito è stato incluso è nota. Questo aiuterà i cambiamenti futuri.
 - Tracciabilità dei requisiti: dipendenze tra i requisiti; dipendenze tra requisiti e architettura e codice del sistema
 - Documentazione dell'architettura del sistema in modo da documentare la struttura del software
 - Refactoring del codice per migliorare la qualità del codice e renderlo più adatto ai cambiamenti

18



UNIVERSITÀ DEGLI STUDI DI SALERNO

19

BASI DI DATI

Metodologie e Modelli per
progetti software



Interazione: Analisi dei requisiti



Giuseppe D'Aniello
Researcher, PhD – IEEE Member

19

BASI DI DATI

Metodologie e modelli per i progetti software

20

ESERCIZIO

Esercizio 4.2 da "Ingegneria del Software, Ian Sommerville, 10ª edizione

➤ **Individuate ambiguità e omissioni nella seguente dichiarazione dei requisiti per una parte del sistema di emissione dei biglietti.**

Un sistema automatico di emissione dei biglietti vende biglietti ferroviari. Gli utenti selezionano la loro destinazione e inseriscono la carta di credito e il codice personale. Il biglietto viene emesso e addebitato sulla carta di credito. Quando l'utente preme il pulsante di avvio, compare un menu che mostra le varie destinazioni possibili, insieme a un messaggio che chiede di selezionare una destinazione e il tipo di biglietto. Una volta selezionata la destinazione, sullo schermo appare il prezzo del biglietto e all'utente è chiesto di inserire la carta di credito. Viene verificata la sua validità, e all'utente viene chiesto di inserire il codice personale di identificazione (PIN). Quando la transazione economica è completata, il biglietto viene emesso.

20

BASI DI DATI

21

Metodologie e modelli per i progetti software

SOLUZIONE

Esercizio 4.2 da "Ingegneria del Software, Ian Sommerville, 10° edizione

➤ **Individuate ambiguità e omissioni nella seguente dichiarazione dei requisiti per una parte del sistema di emissione dei biglietti.**

1. Un cliente può comprare insieme diversi biglietti per una stessa destinazione o deve comprarli uno alla volta?
2. Un cliente può cancellare una richiesta se è stato fatto un errore?
3. Come dovrebbe rispondere il sistema se è stata inserita una carta non valida?
4. Cosa succede se il cliente prova a inserire la carta prima di selezionare una destinazione?
5. L'utente deve premere il pulsante di avvio nuovamente se vogliono comprare un altro biglietto per una destinazione diversa?
6. Il sistema dovrebbe vendere biglietti solo tra la stazione in cui si trova la macchina e le destinazioni dirette o dovrebbe essere possibile includere anche altre destinazioni?

21

BASI DI DATI

22

Metodologie e modelli per i progetti software

ESERCIZIO

➤ **Consideriamo ancora il sistema di emissione dei biglietti. Quali sono i dati che dovrà gestire?**

Un sistema automatico di emissione dei biglietti vende biglietti ferroviari. Gli utenti selezionano la loro destinazione e inseriscono la carta di credito e il codice personale. Il biglietto viene emesso e addebitato sulla carta di credito. Quando l'utente preme il pulsante di avvio, compare un menu che mostra le varie destinazioni possibili, insieme a un messaggio che chiede di selezionare una destinazione e il tipo di biglietto. Una volta selezionata la destinazione, sullo schermo appare il prezzo del biglietto e all'utente è chiesto di inserire la carta di credito. Viene verificata la sua validità, e all'utente viene chiesto di inserire il codice personale di identificazione (PIN). Quando la transazione economica è completata, il biglietto viene emesso.

22

BASI DI DATI

23

Metodologie e modelli per i progetti software

SOLUZIONE

➤ **Consideriamo ancora il sistema di emissione dei biglietti. Quali sono i dati principali che dovrà gestire?**

- Biglietto: data, ora, percorso, costo, validità?, classe?, posto? ...
- Probabilmente dobbiamo considerare l'esistenza di diversi tipi di biglietti: biglietti con una durata massima, biglietto con una distanza massima (e.g. zone), biglietto nominale, biglietto giornaliero...
- Utente: codice personale. La specifica non è chiara su questo punto, ma se esiste un codice personale è probabile siano memorizzati i dati anagrafici dell'utente...
- Pagamento: data, ora, metodo di pagamento, importo...
- Destinazioni: città, provincia, nome della stazione...

Un sistema automatico di emissione dei biglietti vende biglietti ferroviari. Gli utenti selezionano la loro destinazione e inseriscono la carta di credito e il codice personale. Il biglietto viene emesso e addebitato sulla carta di credito. Quando l'utente preme il pulsante di avvio, compare un menu che mostra le varie destinazioni possibili, insieme a un messaggio che chiede di selezionare una destinazione e il tipo di biglietto. Una volta selezionata la destinazione, sullo schermo appare il prezzo del biglietto e all'utente è chiesto di inserire la carta di credito. Viene verificata la sua validità, e all'utente viene chiesto di inserire il codice personale di identificazione (PIN). Quando la transazione economica è completata, il biglietto viene emesso.

23

BASI DI DATI

24

Metodologie e modelli per i progetti software

ESERCIZIO

Esercizio 4.4 da "Ingegneria del Software, Ian Sommerville, 10ª edizione

➤ **Individuate alcuni requisiti non funzionali per il sistema di emissione dei biglietti, specificando l'affidabilità richieste e il tempo di risposta**

24

BASI DI DATI

25

Metodologie e modelli per i progetti software

SOLUZIONE

Esercizio 4.4 da "Ingegneria del Software, Ian Sommerville, 10° edizione

➤ **Individuate alcuni requisiti non funzionali per il sistema di emissione dei biglietti, specificando l'affidabilità richieste e il tempo di risposta**

- Tra le 6:00 e le 23:00 di ogni giorno, il tempo totale di fermo macchina non dovrebbe essere superiore ai 5 minuti
- Tra le 6:00 e le 23:00 di ogni giorno, il tempo di recupero dopo un errore del sistema non dovrebbe essere superiore ai 2 minuti
- Tra le 23:00 e le 6:00 di ogni giorno, il tempo totale di fermo macchina non dovrebbe essere superiore ai 20 minuti

Questo tipo di requisiti vengono definiti anche requisiti di affidabilità.

25

BASI DI DATI

26

Metodologie e modelli per i progetti software

SOLUZIONE

Esercizio 4.4 da "Ingegneria del Software, Ian Sommerville, 10° edizione

➤ **Individuate alcuni requisiti non funzionali per il sistema di emissione dei biglietti, specificando l'affidabilità richieste e il tempo di risposta**

- Dopo che l'utente ha premuto un pulsante, il display dovrebbe essere aggiornato entro 0.5 secondi
- Il tempo di emissione del biglietto dopo che la validazione della carta di credito è stata effettuata non dovrebbe essere superiore ai 5 secondi
- Quando la carta di credito è in fase di validazione, il display dovrebbe mostrare un messaggio al cliente per indicare che il processo è in corso.

26

BASI DI DATI

27

Metodologie e modelli per i progetti software

ESERCIZIO

- Definire alcuni requisiti funzionali e non funzionali del sistema di rilevazione presenze EasyBadge dell'Università di Salerno

27

BASI DI DATI

28

Metodologie e modelli per i progetti software

SOLUZIONE

- Definire alcuni requisiti funzionali e non funzionali del sistema di rilevazione presenze EasyBadge dell'Università di Salerno

FUNZIONALI

- L'utente di tipo docente deve poter avviare la registrazione delle presenze per una lezione precedentemente pianificata.
- L'utente di tipo studente deve poter registrare la presenza alla lezione inserendo il codice statico fornito dal sistema e comunicato dal docente
- L'utente di tipo studente dovrebbe poter registrare la presenza anche utilizzando un QR Code
- ...

NON FUNZIONALI

- Tra le 8:30 e le 18:30 dei giorni feriali il tempo totale di fermo del sistema non deve essere superiore ai 5 minuti
- Il sistema deve essere in grado di gestire almeno 300 studenti che rilevano la presenza contemporaneamente per ogni lezione, con un ritardo nel completamento della procedura di rilevazione delle presenze di massimo 5 secondi
- L'applicazione utilizzata dall'utente studente deve essere compatibile con Android >6.0 e iOS >10.0
- L'applicazione deve supportare la lingua italiana e l'inglese.
- ...

28

BASI DI DATI

29

Metodologie e modelli per i progetti software

ESERCIZIO

➤ **Identificare alcuni dei dati che deve gestire il sistema EasyBadge**

29

BASI DI DATI

30

Metodologie e modelli per i progetti software

SOLUZIONE

➤ **Identificare alcuni dei dati che deve gestire il sistema EasyBadge**

- **Studente:** matricola, dati anagrafici, corso di studio di appartenenza, anno accademico, piano di studi?
- **Docente:** matricola, dati anagrafici, dipartimento di appartenenza, insegnamenti, etc.
- **Lezione:** data, ora, durata, insegnamento, codice statico associato alla lezione, codici dinamici?
- **Insegnamento:** matricola, Corso di Laurea, titolo, descrizione, docenti di riferimento, lezioni?
- **Presenza:** studente, lezione, data, ora, metodo per la rilevazione della presenza...
- ...

30

BASI DI DATI

31

Materiale utilizzato e bibliografia

➤ *Le slide utilizzate dai docenti per le attività frontali sono in gran parte riconducibili e riprese dalle slide originali (con alcuni spunti parziali ripresi dai libri indicati) realizzate da:*

- ✓ *autori del libro Basi di Dati (Atzeni e altri) testo di riferimento del corso Basi di Dati e sono reperibili su internet su molteplici link oltre che laddove indicato dagli stessi autori del libro;*
- ✓ *Prof. Ian Sommerville Ingegneria del Software.*
- ✓ *Prof. Aniello Cimitile per corsi di Ingegneria del Software dell'Università degli Studi del Sannio al seguente link ed altri: <http://www.ing.unisannio.it/cimitile/ingsw1718/>*
- ✓ *Prof. Paolo Ciancarini per corsi di Ingegneria del software dell'Università di Bologna <http://www.cs.unibo.it/~cianca/wwwpages/ids/2.pdf>*

➤ *Diverse slide su specifici argomenti utilizzate dai docenti per le attività frontali sono anche in parte riconducibili e riprese dalle slide originali – facilmente reperibili e accessibili su internet*