



Outline

- Interfaccia con la fase di analisi dei requisiti
- Sviluppo del disegno di sistema
- Interfaccia con la fase di testing
- Interfaccia con la fase di codifica
- Documentations
- Cosa ci dice Pmango
- Thanks



Abbiamo adattato ICONIX all'analisi, non siamo riusciti ad usarlo in modo effettivo nella progettazione

Use case driven

- Esiste una relazione R biunivoca tra la definizione dei metodi definiti nel disegno di sistema con la definizione degli use case
- Semplice tracciare le funzionalita' definite nella specifica dei requisiti sul disegno di sistema (per transitivita' su Ucs) per assicurazione della qualita'
- Raffinamento dei use case e loro formalizzazione usando la notazione dei diagrammi di sequenza
- Vantaggio di avere una specifica abbastanza raffinata delle sequenze di esecuzione



Costruzione del class diagram

- Partenza dal Domain model prodotto dall'analisi
- Raffinamento di concetti poco definiti (user option choice)
- Pulizia di quelli superflui (troppo dettaglio su taskbox)
- Non ben specificato ne le strutture dati, ne i metodi non mappati su use case



La progettazione ha utilizzato un approccio orientato ai componenti

Pros

- cattura delle responsabilita' in componenti, i concetti che variano sono stati ben incapsulati (taskdatatree, gifarea, gen)
- processo iterativo, raffinamento dei componenti costante
- le interazioni sono state ben formalizzate (slide su UC)
- Semplice suddividere del lavoro sulle risorse
- Fornire al team di testing oggetti indipendenti da esercitare



Cons

- sottovalutata l'interazione con l'architettura esistente (data source, interactive choice, reports)
- a livello di astrazione piu basso della definizione delle classi non e' stata data una formalizzazione riguardo:
 - ADT per ogni concetto
 - Relazioni di client e inheritance
 - Interazione tra oggetti



Il modello ICONIX mira ad avere use case ben formati e non ambigui. Per raggiungere questo obiettivo utilizza lo strumento robustness diagram

Soprattutto per il poco tempo a disposizione non abbiamo potuto usarli. Ci avrebbero avvantaggiato su questi aspetti:

- Individuare le tuple di testing (controller → tuple test)
- Specifica delle sequenze di ogni test
- Migliore specifica del comportamento dei componenti e delle loro interazioni



Il disegno di sistema ha formalizzato il lavoro svolto nell'analisi, ma non ha tenuto conto di questi aspetti non in relazione con l'analisi:

- Nessun suggerimento sull'implementazione di codice relativo a argomenti importanti (algoritmi dei generatori)
- Sottostima di alcuni concetti da implementare (critical path, data retrieving, user interaction, reports)
- Specifica dei moduli molto poco dettagliata, lasciando molto lavoro (anche di natura progettuale) alla fase di coding



Abbiamo scelto di non usare il modello proposto in modo da poterci adattare meglio al nostro obiettivo di implementare ICONIX

- Ci e' sfuggita il suggerimento “indicazioni algoritmiche”
- Dimenticata la sezione “Dipendenze esterne e riuso”
- Chiarezza nell'esporre i concetti statici (packages, classes) e dinamici (sequence)



Base Information		Dependencies		Tasks depending on this Task	
Project:	[Kiwi-TP]PMango: Gantt e CPM	none		none	
Task:	2.1 Progettazione				
Task Parent:	Sviluppo				
Start Date:	16/11/2009 09:00 am	Effort:	25 ph		
Finish Date:	23/12/2009 05:00 pm	Budget:	750 €		
Computed Information at 18/10/2009 12:00 am					
Start Date from Tasks:	23/11/2009	Actual Start Date:	25/11/2009		
Finish Date from Tasks:	05/12/2009	Actual Finish Date:	25/12/2009		
Effort from Tasks:	25 ph	Actual Effort:	22 ph		
Budget from Tasks:	750 €	Actual Cost:	660 €		
Progress:	100%	Effort Performance Index:	0.88		
Time Performance Index:	1.05	Cost Performance Index:	0.88		
Assigned to task					
Person:	Role:	Effort:			
Rogai, Nicolò	Software Designer	8 ph			
Calabri, Francesco	Software Designer	8 ph			
Tinacci, Marco	Software Designer	9 ph			
Details		Properties			
Status:	Active	Well Formed: <input type="checkbox"/> Task is Defined			
Priority:	normal	Cost Effective: <input type="checkbox"/> Task is Well Formed			
URL:		Effort Effective: <input type="checkbox"/> Task is Cost Effective			
Description:		Time Effective: <input type="checkbox"/> Task isn't Time Effective			
		- Task time performance index is greater than 1			
		<input type="button" value="compute"/>			

- Pianificazione temporale stretta, la fase di testing si e' sviluppata all'interno della progettazione (start_date(testing) = 01/12/2009, end_date(testing) = 09/12/2009 entrambe actual)
- Stima dello sforzo corretta, non contando i cons
- Impiego delle risorse sotto il budget previsto
- I backup astraggono sulle ripianificazioni



Thank you Kiwi team
Thank you Giovanni