Presentazione Definizione di Prodotto per C04

Egoless Group

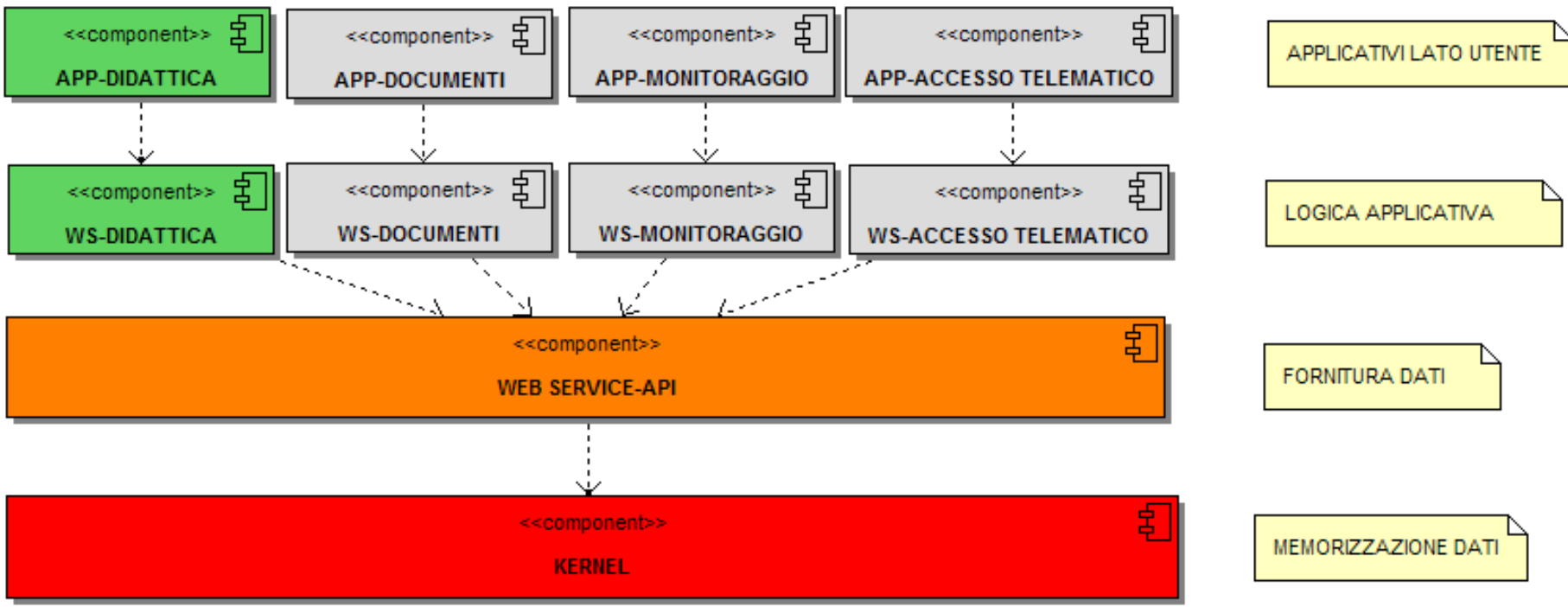
Revisione di Progetto Definitivo
28 febbraio 2007

Outline

Pianificazione

- le ore erogate sono molto di più di quelle previste
- di conseguenza la pianificazione è slittata in avanti
- stiamo lavorando per ripianificare in maniera adeguata

Architettura generale



Outline

Tecnologie usate per l'implementazione

- Java Enterprise Edition 5, in particolare
  - Glassfish, application server Java EE
  - JAX WS 2.0 per creare il Web Service

Dettaglio di WS Didattica

- classi generate da JAX WS per comunicare con WEB SERVICE API
- classe Enterprise JavaBean che implementa la logica di WS Didattica
- ogni metodo esposto allo strato applicativo lancia un'eccezione quando si verificano errori

Outline

Tecnologie usate

- Java 5, in particolare
  - Swing
  - JavaBean
  - . . . tutto con l'ausilio di NetBeans 5.5 con Enterprise Pack

Dettaglio di APP Didattica

- Applicazione strutturata secondo il pattern architetturale MVC
- ogni classe da manipolare viene gestita da un modello implementato usando JavaBean:
  - recuperare oggetti dal Web Service sottostante
  - permettere l'accesso e/o modifica delle proprietà dell'oggetto
  - notificare le modifiche a tutte le View in ascolto (. . . pattern Observer)
- l'interfaccia grafica in Swing fa contemporaneamente da View e da Controller

Evoluzione del Piano di Qualifica

- Definizione dell'infrastuttura tecnologica
  - Strumenti di test
  - Strumenti di analisi statica
  - Tool per il tracciamento
- Adozione di metriche
  - per la progettazione
  - per la codifica

Definizione dell'infrastruttura tecnologica

Strumenti di analisi

- Strumenti di test
  - JUnit per il test funzionale
  - Cobertura per la verifica di branch e statement coverage
- Strumenti di analisi statica
  - FindBugs per ricerca di bug patterns
- altri (ricerca in corso)

Definizione dell'infrastruttura tecnologica

Tool per il tracciamento

- utilizzeremo OSRMT (www.osrmt.com)
- infrastruttura semplice da creare (file database condiviso con svn)
- rapido da introdurre
  - 2 ore per studiare il tool
  - 2 ore per introdurre i dati

Metriche adottate

- Fan-in e Fan-out
  - Cercheremo un tool per la misurazione automatica
- SLOC
  - porre (eventualmente) limiti alla lunghezza dei metodi o dei file
- Complessità ciclomatica
  - Posti dei limiti iniziali, da adattare se necessario
  - Misurata automaticamente (Cobertura)
- Altre in fase di studio (Function Point, ...)

Cosa abbiamo sbagliato finora

- pianificazione
  - aumento del numero di ore progettazione per acquisire padronanza delle tecnologie adottate
- adottato BoUML al posto di ArgoUML
  - supporto a UML 2.0
  - meno user-friendly ma più completo
- collaborazione in fase di assestamento