



UNIVERSITA' DEGLI STUDI DI SALERNO

LAUREA MAGISTRALE IN TECNOLOGIE INFORMATICHE E MANAGEMENT

CORSO DI INGEGNERIA DEL SOFTWARE: MANUTENZIONE E TESTING

Gestione del Portale dedicato al Dottorato di Ricerca in Informatica

Documento di analisi

di manutenzione

18/10/2015

Versione 1.0

Coordinatori del progetto

Prof. Andrea De Lucia

Partecipanti

D'Eugenio Elisa - Matricola 0222500122

Revision History

Data	Versione	Descrizione	Autori
18/10/2015	1.0	Prima stesura	D'Eugenio Elisa

Indice

1. Scopo del documento
2. Panoramica del sistema attuale
 - 2.1 Attori e funzionalità
 - 2.2 Design, implementazione e testing del sistema attuale
 - 2.2.1 Design
 - 2.2.2 Implementazione
 - 2.2.3 Testing
3. Analisi della modifica richiesta
4. Individuazione della soluzione progettuale
 - 4.1 Problematiche affrontate
 - 4.2 Soluzione individuata
 - 4.3 Soluzioni alternative
5. Identificazione dell'Impact Set
6. Studio di fattibilità
 - 6.1 Identificazione, descrizione e valutazione dei costi
 - 6.2 Identificazione, descrizione e classificazione dei benefici
 - 6.3 Identificazione, descrizione e classificazione dei rischi
 - 6.4 Risultati previsti

Scopo del documento

In questo documento saranno descritti gli obiettivi del processo di manutenzione del sistema Phd-platform, in riferimento al documento di “Identificazione e classificazione delle modifiche richieste”.

Verrà studiato come queste possano impattare sugli artefatti del sistema esistente, analizzando il rapporto costi/benefici e i possibili rischi derivanti dalla fase di progettazione.

Questo documento includerà uno studio di fattibilità e verranno pianificate le fasi successive di progettazione, implementazione e testing.

Panoramica del sistema attuale

Il sistema realizzato consiste in un portale web per la gestione dei cicli di dottorato, afferenti al Dipartimento Studi e Ricerche Aziendali dell’Università di Salerno. Il sistema consente di avere tutte le informazioni relative ai cicli di dottorato attivi, quindi ai curriculum di dottorato presenti in ogni ciclo, al collegio docenti e ai dottorandi che vi fanno parte.

Il sistema fa parte di un portale più ampio relativo al Dipartimento di Studi e Ricerche Aziendali, contiene quindi elementi esterni di altre sezioni del portale.

Attori e funzionalità

Il sistema prevede l’esistenza di tre attori principali:

- Amministratore
- Professore
- Dottorando

Tuttavia, essendo parte di un portale più ampio, il sistema dedicato al Dottorato del dipartimento DISTRA deve tener conto di altri attori presenti nel sistema, anche se non sono di interesse per il sistema oggetto della manutenzione.

Design, implementazione e tesng del sistema attuale

Design

Il sistema Phd-platform è stato progettato e implementato utilizzando lo schema three-tier per suddividere il sistema in tre sottosistemi, ognuno dei quali fornisce servizi specifici.

Interface Layer:

Include tutte le interfacce grafiche e in generale i boundary objects con cui interagisce l'utente.

Application Layer:

Include tutti gli oggetti relativi al controllo e alle entità del sistema.

Storage Layer:

Effettua la memorizzazione, il recupero e l'interrogazione degli oggetti persistenti.

Il sistema software è stato sviluppato con il linguaggio di programmazione object oriented Java™, coniugato all'apporto delle librerie JDBC per la connessione al DBMS.

L'application layer è stato sviluppato con le tecnologie JSP e Servlet.

Secondo le change request presentate, l'architettura non subirà modifiche in quanto non sono ritenute necessarie.

Implementazione

L'implementazione del software è stata effettuata in riferimento al modello descritto in precedenza, in particolare associando ad ogni layer un insieme di classi che sviluppano delle funzionalità richieste nella fase di analisi dei requisiti.

Nello stile architetturale del software Phd-platform il presentation layer è caratterizzato dalle interfacce grafiche utente:

- l'interfaccia amministratore;

- l'interfaccia docente;
- interfaccia studente.

Sono presenti ulteriori interfacce che non interessano il dottorato di ricerca.

L'application layer contiene tre sottosistemi:

- gestione ciclo;
- gestione curriculum;
- gestione classe;
- gestione account;
- gestione autenticazione e iscrizione;
- gestione tutorato.

Sono presenti ulteriori gestioni che non interessano il dottorato di ricerca.

Lo storage layer contiene la gestione dei dati persistenti e comunica con l'application layer attraverso il framework JDBC.

Testing

Il testing effettuato sul precedente sistema Phd-platform è stato effettuato tramite Unit testing: la tecnica scelta riferisce alla metodologia black box, ovvero uno sviluppo incentrato sul controllo dell'input e dell'output previsti. La procedura inizia con l'individuazione delle classi di equivalenza, mediante le quali è possibile determinare il partizionamento di tutti i possibili input.

Analisi della modifica richiesta

Il processo di manutenzione prevede che i requisiti funzionali esistenti rimangano invariati, mentre gli attori che interagiscono con il sistema dovranno essere rivisti in modo da eliminare quelli che interessano il sistema più ampio del portale relativo al DISTRA, ma che non interessano il dottorato.

Gli obiettivi principali consistono nella realizzazione delle seguenti modifiche:

- **Correzione di errori**: il sistema esistente non ha soddisfatto a piano i requisiti degli stakeholder in termini di layout e organizzazione delle informazioni all'interno del portale. Per questo motivo, si rende necessario un ulteriore incontro con gli stakeholder al fine di analizzare le loro richieste e progettare un nuovo layout.
- **Adattamento database**: il sistema voluto, non dovendo essere parte di un portale più ampio (portale del Dipartimento di Studi & Ricerche Aziendali), potrebbe essere opportuno adattare quella parte del database relativa alla gestione utente in quanto attualmente contiene informazioni non utili al sistema preso in considerazione. Questo comporterebbe una modifica sostanziale anche del codice relativo al portale, oltre che del database.
- **Aggiunta della funzionalità "Calendario"**: il sistema esistente ha lo scopo di facilitare la gestione delle attività di dottorato. A tal fine, si vuole aggiungere una nuova funzionalità: la gestione del calendario. Questa gestione si riferisce all'inserimento, la modifica, la cancellazione e la visualizzazione di corsi e seminari relativi ad determinati cicli di dottorato o curriculum. In questo modo i dottorandi possono sempre essere aggiornati sui loro impegni accademici.
- **Aggiunta della funzionalità "Presenze"**: il sistema esistente ha lo scopo di facilitare la gestione delle attività di dottorato. A tal fine, si vuole aggiungere una nuova funzionalità, legata alla funzionalità "calendario": la gestione delle presenze. In questo modo si potrebbe tener traccia, nel portale, delle presenze e delle assenze relative ai corsi e ai seminari. Questa attività viene attualmente svolta in formato cartaceo.
- **Aggiunta della funzionalità "News e Avvisi"**: il sistema esistente ha lo scopo di facilitare la gestione delle attività di dottorato. A tal fine, si vuole aggiungere una nuova funzionalità: la gestione di news e avvisi. In questo modo sarà possibile aggiornare gli utenti del portale direttamente sugli avvisi e le news di interesse.

Individuazione della soluzione progettuale

Problematiche affrontate

Issue 1: Quali funzionalità verranno mantenute nel nuovo sistema?

Proposal 1.1
Mantenere tutte le funzionalità del sistema.
Proposal 1.2
Mantenere le funzionalità del sottosistema relativo al dottorato di ricerca.

Criterion 1.1
Scegliere funzionalità pertinenti con il sistema da realizzare.
Criterion 1.2
Scegliere funzionalità che migliorano la manutenzione futura.

Argument 1.1
Mantenere tutte le funzionalità del vecchio sistema rende la manutenzione più complessa in quanto le funzionalità che non riguardano il nuovo sistema sono nettamente maggiori rispetto a quelle necessarie. Inoltre richiedono di gestire una notevole quantità di dati non necessari.
Argument 1.2
Mantenere solo le funzionalità del sottosistema relativo al dottorato di ricerca richiede tempo e attenzione maggiore, in quanto non esiste una netta separazione tra sottosistemi presenti nel sistema precedente. In questo modo, però, si eliminano informazioni inutili e si rende più facile la manutenzione futura.

Resolution 1.1

Analizzando le argomentazioni Argument 1.1 e Argument 1.2, l'Issue 1 è stata risolta applicando la Proposal 1.2.

Issue 2: Sono necessarie nuove funzionalità?

Proposal 2.1

Nessuna nuova funzionalità aggiunta.

Proposal 2.2

Aggiunta di nuove funzionalità per la consultazione di un calendario dei corsi, delle presenze relative ad una lezione e delle news relative al dottorato.

Proposal 2.3

Aggiunta di nuove funzionalità social, come lo scambio di email e la possibilità di commentare/condividere degli elaborati.

Criterion 2.1

Sviluppare funzionalità solo se strettamente necessario per ridurre i tempi di sviluppo.

Criterion 2.2

Sviluppare funzionalità in linea con i requisiti definiti dagli stakeholder.

Argument 2.1

Le funzionalità attuali consentono l'utilizzo del sistema. Nuove funzionalità non sono vitali.

Argument 2.2

Le funzionalità relative al calendario, alle presenze e alle news sono espressamente richieste dagli stakeholder. Tali funzionalità sono state previste nel vecchio sottosistema, ma non sono state implementate.

La disposizione del database consente di inserire le nuove funzionalità in tempi brevi.

Argument 2.3

Le funzionalità social sono espressamente richieste dagli utenti del sistema. Tali funzionalità non sono state previste nel vecchio sottosistema.

La disposizione del database non consente di inserire le nuove funzionalità in tempi brevi.

Resolution 2.1

Analizzando le argomentazioni Argument 2.1, Argument 2.2 e Argument 2.3, l'Issue 2 è stata risolta applicando la Proposal 2.2.

Issue 3: Quali, tra le funzionalità aggiuntive, hanno priorità maggiore?

Proposal 3.1

Gestione Calendario.

Proposal 3.2
Gestione Presenze.
Proposal 3.3
Gestione Avvisi.

Criterion 3.1
Sviluppare le funzionalità più richieste dagli stakeholder.
Criterion 3.2
Sviluppare le funzionalità più semplici e che richiedono meno tempo.

Argument 3.1
La gestione calendario è tra le più utili e richieste dagli stakeholder, il tempo richiesto per la realizzazione è medio.
Argument 3.2
La gestione calendario è tra le più utili e richieste dagli stakeholder, il tempo richiesto per la realizzazione è medio ma non semplice in quanto c'è bisogno di attendere che la gestione calendario sia realizzata.
Argument 3.3
La gestione calendario non è tra le più utili e richieste dagli stakeholder, ma il tempo richiesto per la realizzazione è breve.

Resolution 3.1

Analizzando le argomentazioni Argument 3.1, Argument 3.2 e Argument 3.3, l'Issue 3 è stata risolta applicando le Proposal 3.1 e Proposal 3.2.

Issue 4: E' necessario riorganizzare il layout?

Proposal 4.1

Nessuna modifica apportata al layout.

Proposal 4.2

Modifica del layout in base alle esigenze dello stakeholder, soprattutto per quanto riguarda la home e la ricerca utente.

Criterion 4.1

Effettuare le modifiche solo se strettamente necessarie per ridurre i tempi di sviluppo.

Argument 4.1

Il layout attuale non soddisfa gli stakeholder.

Argument 4.2

Le modifiche del layout sono fondamentali per l'accettazione da parte degli stakeholder.

Resolution 4.1

Analizzando le argomentazioni Argument 4.1 e Argument 4.2, l'Issue 4 è stata risolta applicando la Proposal 4.2.

Issue 5: E' necessario riorganizzare il database del sistema?

Proposal 5.1

Mantenere il database esistente.

Proposal 5.2

Mantenere solo la parte di database necessaria per la gestione del dottorato.

Criterion 5.1

Mantenere parte di database pertinente con il sistema da realizzare.

Criterion 5.2

Mantenere parte di database che migliora la manutenzione futura.

Argument 5.1

Mantenere il database esistente rende la manutenzione futura più complessa in quanto le

funzionalità che non riguardano il nuovo sistema sono nettamente maggiori rispetto a quelle necessarie. Inoltre richiedono di gestire una notevole quantità di dati non necessari.

Argument 1.2

Mantenere solo la parte di database necessaria per la gestione del dottorato. richiede tempo e attenzione maggiore, in quanto non esiste una netta separazione tra sottosistemi presenti nel sistema precedente. In questo modo, però, si eliminano informazioni inutili e si rende più facile la manutenzione futura.

Resolution 5.1

Analizzando le argomentazioni Argument 5.1 e Argument 5.2, l'Issue 5 è stata risolta applicando la Proposal 5.2.

Issue 6: Che tipologia di linguaggi di programmazione utilizzare per lo sviluppo del nuovo sistema?

Proposal 6.1

Per l'implementazione dei moduli lato client si utilizza HTML, CSS e Javascript, mentre per l'implementazione dei moduli lato server si utilizza il linguaggio Java per applicazioni Web based: JSP e Servlet Java.

Proposal 6.2

Per l'implementazione dei moduli lato client si utilizza HTML, CSS e Javascript, mentre per l'implementazione dei moduli lato server si utilizza il linguaggio di programmazione PHP.

Criterion 6.1

L'implementazione dei moduli client e server deve essere funzionante indipendentemente dalla pia'aforma.

Criterion 6.2

L'implementazione deve essere facilitata dal riuso del codice.

Argument 6.1

Utilizzando la combinazione HTML CSS e Javascript per l'implementazione dei moduli lato client, si assicura il funzionamento su ogni browser compatibile, indipendentemente dalla piattaforma software/hardware. Per questi moduli è possibile il riuso di codice preesistente poiché tali linguaggi sono presenti nella progettazione del sistema esistente.

I moduli del server sono implementati utilizzando Java Servlet e pagine JSP, che consentono di eseguire classi Java e/o script Java su richiesta del client. E' possibile il riuso del codice esistente nel sistema attuale poiché sviluppato nello stesso linguaggio di programmazione.

Argument 6.2

Utilizzando la combinazione HTML CSS e Javascript per l'implementazione dei moduli lato client, si assicura il funzionamento su ogni browser compatibile, indipendentemente dalla piattaforma software/hardware. Per questi moduli è possibile il riuso di codice preesistente poiché tali linguaggi sono presenti nella progettazione del sistema esistente.

I moduli del server sono implementati utilizzando il linguaggio PHP, che consente di eseguire classi PHP su richiesta del client. In questo caso si dovrebbe attuare il porting del codice esistente in PHP o sviluppare un wrapper.

Resolution 6.1

Analizzando le argomentazioni Argument 6.1 e Argument 6.2, l'Issue 6 è stata risolta applicando la Proposal 6.1.

Soluzione individuata

La soluzione individuata consiste nell'unione di tutte le “resolution” ottenute nel paragrafo “Problematiche affrontate”. In dettaglio, la soluzione individuata dovrà avere le seguenti caratteristiche:

- Mantenere le funzionalità del sottosistema relativo al dottorato di ricerca;
- Aggiunta di nuove funzionalità per la consultazione di un calendario dei corsi, delle presenze relative ad una lezione e delle news relative al dottorato;
- Priorità alte per la realizzazione delle nuove gestioni Calendario e Avvisi. Minore priorità per la gestione Presenze.
- Modifica del layout in base alle esigenze dello stakeholder, soprattutto per quanto riguarda la home e la ricerca utente;
- Mantenere solo la parte di database necessaria per la gestione del dottorato;
- Per l'implementazione dei moduli lato client si utilizza HTML, CSS e Javascript, mentre per l'implementazione dei moduli lato server si utilizza il linguaggio Java per applicazioni Web based: JSP e Servlet Java.

Soluzioni alternative

Le soluzioni alternative sono rappresentate da tutte le proposte e argomentazioni affrontate nel paragrafo “Problematiche affrontate”, ma che non sono state selezionate per la soluzione adottata.

Identificazione dell'Impact Set

La soluzione individuata ha coinvolto diversi artefatti del sistema attuale. La tabella che segue descrive il Candidate Impact Set individuato, ovvero l'insieme di tutti gli artefatti che verranno modificati, durante la fase di manutenzione. Inoltre, accanto ad ogni artefatto sarà associato un livello di impatto, ovvero tale livello farà riferimento al modo in cui la modifica va ad impattare sul sistema software attuale.

Per individuare le componenti del sistema impattate dalla modifica si utilizzerà un approccio topdown, ovvero a partire dai documenti di alto livello si andrà ad individuare gli artefatti di più basso livello che andranno ad essere modificati, durante la fase di manutenzione.

L'interfaccia web deve consentire agli utenti di effettuare le stesse operazioni che venivano compiute nel sistema attuale; per questa ragione non ci si aspettano cambiamenti nei requisiti funzionali e nei casi d'uso già presenti nel documento di analisi e specifica dei requisiti.

Per la stessa ragione, poiché il nuovo modulo di Presentation andrà a ricreare gli stessi oggetti già presenti nel livello di Presentation precedente, non ci si aspettano cambiamenti nel modello dinamico dell'applicazione.

In definitiva, nel documento di analisi e specifica dei requisiti dovranno solo essere aggiunti i requisiti funzionali e i casi d'uso che riguardano le funzionalità che saranno aggiunte, ovvero la funzionalità di gestione del calendario, delle presenze e degli avvisi.

E' importante sottolineare che tale documento non risulta completo, in quanto sono attualmente presenti soltanto alcune sezioni del documento come: requisiti funzionali, casi d'uso, diagrammi dei casi d'uso e mockup.

Per quanto riguarda i documenti SDD e ODD, non ci sono modifiche in quanto i documenti non sono stati prodotti per il sistema precedente. Bisogna quindi effettuare una loro prima stesura.

Studio di fattibilità

Identificazione, descrizione e valutazione dei costi

Nelle tabelle che seguono sono indicati gli artefatti che saranno impattati. L'impatto della modifica verrà valutato utilizzando tre categorie:

- FORTE: se saranno necessarie pesanti modifiche nell'artefatto o se l'artefatto dovrà essere completamente sostituito;
- MEDIO: se saranno necessarie sostanziali modifiche all'artefatto, non facendo cambiare però la sua struttura in maniera eccessiva;
- DEBOLE: se saranno necessarie solo modifiche marginali;

Identificazione	Valutazione	Motivazioni
Mantenere le funzionalità del sottosistema relativo al dottorato di ricerca.	DEBOLE	La manutenzione del sistema verrà effettuato in modo incrementale. Pertanto Il un primo momento si effettueranno solo le modifiche necessarie affinché le funzionalità precedentemente implementate siano funzionanti dopo la separazione del sottosistema dal sistema precedente.
Aggiunta di nuove funzionalità.	MEDIA	La modifica richiesta prevede l'aggiunta della funzionalità per la consultazione di un calendario dei corsi, delle presenze relative ad una lezione e delle news relative al dottorato. Tali funzionalità possono essere implementate a partire da elementi già presenti nel vecchio

		sistema.
Modifica del layout.	MEDIA	Le modifiche del layout consistono principalmente nell'includere alcune interfacce nella home del sistema, per una consultazione più rapida e completa del portale.
Mantenere solo la parte di database necessaria per la gestione del dottorato;	MEDIA	La manutenzione del sistema verrà effettuato in modo incrementale. Pertanto Il un primo momento si effettueranno solo le modifiche necessarie affinché le funzionalità precedentemente implementate siano funzionanti dopo la separazione del sottosistema dal sistema precedente.
Implementazione con linguaggi di programmazione client-side HTML, CSS, JavaScript e serverside Java.	FORTE	La maggior parte del sistema è implementata tramite l'utilizzo di tali tecnologie. Purtroppo il team di sviluppo non ha esperienza in tali tecnologie.
Gestione della sicurezza e della riservatezza.	DEBOLE	La gestione della sicurezza è realizzata tramite autenticazione per mezzo di email e password, mentre la gestione della riservatezza non è prevista nelle modifiche richieste, né tantomeno nei requisiti del sistema attuale.
Organizzazione del lavoro nel team di sviluppo.	MEDIO	I componenti del team di sviluppo non hanno esperienza in progetti simili. Pertanto verranno effettuate due

		riunioni a settimana.
Testing funzionale	FORTE	I requisiti funzionali testati nel sistema attuale sono pochi e coprono pochi casi di test. Pertanto, il testing verrà affrontato ex-novo.
Testing di accettazione	FORTE	E' necessaria interazione con il cliente finale per analizzare i risultati del test.

Identificazione, descrizione e classificazione dei benefici

Identificazione	Valutazione	Motivazioni
Facilità di distribuzione e manutenzione.	FORTE	L'applicazione web si trova interamente sul server, per cui la pubblicazione o l'aggiornamento di risorse sul nodo server è automaticamente reso disponibile a tutti i nodi client.
Accesso multiplatforma.	FORTE	L'accesso all'applicazione web è indipendente dall'hardware e dal sistema operativo utilizzato dagli utenti.
Riduzione del costo di gestione.	FORTE	L'uso di Internet come infrastruttura per un'applicazione web riduce notevolmente sia i costi di connettività

		che i costi di gestione dei client.
Scalabilità.	FORTE	Il nuovo sistema software può crescere insieme alle esigenze del cliente senza particolari problemi.

Identificazione, descrizione e classificazione dei rischi

Identificazione	Valutazione	Motivazioni
Introduzione di nuovi fault.	MEDIA	Il testing che si intende effettuare aumenta le probabilità di riscontrare nuovi errori.
Malfunzionamenti del Server.	DEBOLE	Malfunzionamenti del Server possono interrompere drasticamente l'interazione con i nodi client.
Attacchi informatici.	DEBOLE	L'uso del web aumenta la probabilità di essere esposti ad attacchi informatici.
Carico di lavoro eccessivo per il Server.	DEBOLE	I client che comunicano con il server potrebbero sbilanciare il carico di lavoro. Tuttavia, questo è un rischio marginale visto il bacino di utenza del software previsto.

Risultati previsti

La realizzazione del sistema software è definita secondo la soluzione individuata nel paragrafo “Soluzione identificata”, tenendo in considerazione costi, benefici e rischi di cui al paragrafo “Studio di fattibilità”.