

Allegato Tecnico 2

Standard tecnologici e architettonici

Evoluzione del Sistema Informativo della CSEA



Allegato Tecnico 2	
Standard tecnologici e architeturali	
Evoluzione del Sistema Informativo della CSEA	

Sommario

1	Scopo del Documento.....	5
2	Le fasi di Progetto	5
2.1	Raccolta esigenze utente	5
2.2	Definizione dei Requisiti.....	5
2.3	Analisi e Disegno	5
2.4	Realizzazione	5
2.5	Collaudo e Verifica Prestazionale.....	5
2.6	Rilascio in esercizio.....	5
3	Piano di Lavoro per i servizi a carattere progettuale.....	6
4	Specifiche requisiti	6
5	Analisi.....	6
6	Disegno architettuale	7
7	Prototipo	8
8	Disegno di dettaglio	8
9	Convalida della tecnologia	9
10	Documentazione Dati	9
11	Modello dei Dati	10
11.1	Dizionario Dati	11
12	Codice sorgente	11
13	Documentazione utente	12
14	Documento di Collaudo	12
15	Manuale di gestione esercizio	13
16	CMDB	13
17	Capacity planning.....	14
18	Piano di Test.....	15
18.1	Categorie di Test.....	16
18.1.1	Test unitari	16
18.1.2	Test di integrazione.....	16
18.1.3	Test di sistema	16



Allegato Tecnico 2	
Standard tecnologici e architeturali	
Evoluzione del Sistema Informativo della CSEA	

18.1.4	Test di accettazione	17
18.2	Aspetti da testare	17
18.2.1	Test funzionali e non funzionali	17
18.2.2	Test di usabilità	17
18.2.3	Test prestazionale	17
18.2.4	Test di stress	17
18.2.5	Test di installazione.....	17
18.3	Modalità di test	18
18.3.1	Test black box	18
18.3.2	Test white box.....	18
18.3.3	Test di regressione	18
19	Riepilogo della documentazione tecnica prevista tra i delivery di progetto.....	19
20	Ambiente di Sviluppo	20
20.1	Riferimenti alle versioni.....	20
21	Documentazione	21
21.1	Javadoc	21
21.2	Convenzioni sulla documentazione	22
22	Codifica.....	23
22.1	Standard per i metodi.....	23
22.1.1	Convenzioni sui nomi (ad esclusione dei costruttori).....	23
22.1.2	Visibilità.....	23
22.2	Standard per gli attributi	23
22.2.1	Convenzione sui nomi.....	23
22.2.2	Visibilità.....	23
22.2.3	Convenzioni sulla documentazione	24
22.3	Standard per i parametri	24
22.3.1	Convenzione sui nomi.....	24
22.3.2	Convenzioni sulla documentazione	24
22.4	Standard per le variabili locali	24
22.4.1	Convenzione sui nomi.....	24



Allegato Tecnico 2	
Standard tecnologici e architettonici	
Evoluzione del Sistema Informativo della CSEA	

22.4.2	Convenzioni sulla dichiarazione variabili e sulla documentazione.....	24
22.5	Standard per le classi.....	24
22.5.1	Convenzione sui nomi.....	24
22.5.2	Convenzione sulla documentazione.....	24
22.5.3	Convenzione sulla dichiarazione.....	25
22.6	Standard per le interfacce.....	25
22.6.1	Convenzione sui nomi.....	25
22.6.2	Convenzione sulla documentazione.....	25
22.7	Standard per i packages.....	25
22.7.1	Convenzione sui nomi.....	25
22.7.2	Convenzione sulla documentazione.....	25
22.7.3	Organizzazione del codice sorgente.....	26
23	Prospetti del modello applicativo e processo di sviluppo.....	27



Allegato Tecnico 2	
Standard tecnologici e architettonici	
Evoluzione del Sistema Informativo della CSEA	

1 Scopo del Documento

Il presente documento illustra gli standard tecnologici ed operativi in essere presso CSEA, una panoramica sui processi di sviluppo software e un riferimento al modello infrastrutturale di base per gli applicativi, che si applica sia allo sviluppo di nuovi Progetti che alla manutenzione del portafoglio Progetti CSEA.

2 Le fasi di Progetto

2.1 Raccolta esigenze utente

La fase di raccolta delle esigenze dell'utente rappresenta il momento in cui vengono valutate le esigenze, anche in termini di urgenza ed importanza.

2.2 Definizione dei Requisiti

La fase di Definizione è volta ad identificare e dettagliare le necessità dell'utente, con riferimento ai processi e alle funzioni che le compongono, al fine di giungere alla definizione dell'ipotesi di soluzione, alla macro pianificazione dei tempi di realizzazione ed al dettaglio dell'effort nella stima di progetto.

2.3 Analisi e Disegno

L'Analisi è volta a definire, in modo completo ed esaustivo, l'applicazione e/o le funzioni da realizzare e/o modificare, con riferimento ai processi individuati e alle modalità con cui tali processi risulteranno visibili all'utente.

Il Disegno è volto a tradurre tutte le caratteristiche della soluzione in specifiche tecniche di dettaglio necessarie alla generazione dei prodotti finali.

2.4 Realizzazione

La fase di Realizzazione è volta a generare i componenti software e le basi dati necessarie alla efficace ed efficiente operatività del sistema oggetto di sviluppo.

2.5 Collaudo e Verifica Prestazionale

Il Collaudo del software realizzato viene effettuato da CSEA, con l'eventuale supporto di terze parti da essa autorizzate, e si conclude con l'accettazione del software.

2.6 Rilascio in esercizio

Il rilascio in esercizio viene effettuato garantendo quanto possibile la continuità ai servizi erogati sia internamente che esternamente.



Allegato Tecnico 2	
Standard tecnologici e architettonici	
Evoluzione del Sistema Informativo della CSEA	

3 Piano di Lavoro per i servizi a carattere progettuale

Relativamente alla realizzazione di un Progetto viene redatto un piano di lavoro che riporta:

- o descrizione e classe di rischio e relativo stato (sospeso, cancellato, ecc.);
- o elenco delle fasi e delle singole attività con relative date di inizio e fine, previste;
- o *deliverables* delle singole fasi e relative date di consegna, previste ed effettive;
- o impegno stimato, secondo la metrica applicabile dell'effort progettuale, ove applicabile suddiviso per fase/attività e per figura professionale;
- o vincoli/ criticità e relative azioni da intraprendere;
- o un gantt delle attività.

Ove necessario:

- o percentuale di avanzamento delle singole attività e data effettiva di chiusura;
- o vincoli/ criticità e relative azioni da intraprendere e/o intraprese;
- o razionali di ripianificazione, scostamento eventuale delle date, dell'impegno e del volume. Il piano dovrà essere corredato del relativo Report attività e risorse.

Deve anche essere presentato un piano di lavoro riepilogativo per tutte le attività in corso, al fine di consentire a CSEA una visione di sintesi delle attività previste per ogni mese in apertura. Tale piano potrà essere prodotto aggregando le informazioni principali dei piani delle singole attività con particolare riferimento alla presenza di un Gantt complessivo.

4 Specifiche requisiti

Il documento di formalizzazione dei requisiti deve contenere la descrizione dei requisiti, funzionali e non, emersi nella fase di definizione delle esigenze utente. Qualora non sia possibile la realizzazione del prototipo eventualmente richiesto, nel documento di specifica dei requisiti deve essere formalizzato il motivo di tale impossibilità.

5 Analisi

Contiene in modo completo ed esaustivo l'analisi dell'applicazione interessata sia relativamente ai processi ed alle modalità con cui tali processi risulteranno visibili agli utenti finali, sia al disegno logico dei dati secondo il modello usato presso CSEA (a oggetti e relazionale), sia per quanto riguarda gli aspetti non funzionali (architettura, sicurezza, accessibilità, vincoli, prestazioni, ecc.), sia alla documentazione delle interfacce (includere



Allegato Tecnico 2	
Standard tecnologici e architetture	
Evoluzione del Sistema Informativo della CSEA	

esempi di layout delle principali schermate utente), sia nei casi in cui è previsto l'utilizzo di un prototipo. Il livello di completezza richiesto deve essere tale da:

- consentire l'approvazione delle funzionalità da parte dei referenti CSEA;
- consentire la produzione del Piano di test senza necessità di ulteriori approfondimenti;
- consentire lo svolgimento della successiva fase di disegno di dettaglio;
- consentire la verifica della stima del volume di software da sviluppare e/o da modificare;
- garantire la tracciabilità con quanto descritto nel documento di requisiti.

6 Disegno architetture

Il documento contiene generalmente la documentazione (UML) sia delle soluzioni architetture in essere sia come output all'attività a supporto dei progetti di sviluppo, contestualizzato nell'architettura dell'ambiente di produzione.

Vengono qui esplicitati eventualmente pattern e framework concordati con CSEA.

Per la progettazione della architettura si parte con l'identificazione dei Fattori Architetture. ovvero dai requisiti (in genere "non funzionali") che condizionano le scelte Architetture.

Alcuni esempi di possibili fattori architetture sono: "possibilità di cambiare il DBMS riducendo al minimo le modifiche necessarie sul codice sorgente", "possibilità di cambiare l'interfaccia grafica riducendo al minimo le modifiche necessarie sul codice sorgente", "inseribilità di strategie di navigazione con approccio data-driven" ..).

Ogni Fattore Architetture deve essere soddisfatto da una decisione (o soluzione) architetture che deve essere prima identificata e successivamente dettagliata. La soluzione architetture può consistere nell'utilizzo di un pattern, o di un componente (libreria o framework).

L'architettura consiste nell'insieme delle decisioni architetture descritte nei promemoria tecnici. Nella logical view tali decisioni sono rappresentate in forma di diagramma mostrando i singoli sottosistemi e le relazioni tra gli stessi.



Allegato Tecnico 2	
Standard tecnologici e architeturali	
Evoluzione del Sistema Informativo della CSEA	

7 Prototipo

Il prototipo, se ritenuto necessario ed anche in accordo con CSEA, è rivolto alla esplicitazione dell'interfaccia utente, in termini di layout e di modalità di utilizzo dell'applicazione. In tal caso la documentazione delle interfacce prevista nel documento Specifiche funzionali potrebbe riportare la sola stampa delle videate del prototipo. Tale prototipazione deve comprendere almeno:

- il layout delle interfacce di colloquio;
- il percorso di navigazione.

Lo strumento di realizzazione del prototipo può differire dagli strumenti che verranno utilizzati per la realizzazione del sistema. La prototipazione deve poter consentire:

- l'eliminazione di eventuali dubbi di fattibilità dell'Obiettivo;
- una migliore comprensione dei requisiti;
- un eventuale test di sistema, nella sua interezza.

Nelle fasi iniziali del ciclo di analisi del Sistema, se ritenuto necessario ed anche in accordo con CSEA, il prototipo potrà essere realizzato anche come *mockup*, sia con tool specialistici che con tool di produzione office, e sarà finalizzato a condividere con l'utente finale il "look-and-feel" delle pagine che saranno realizzate e i percorsi di navigazione.

Nella fase di Disegno, il prototipo dovrà invece evolvere e consentire all'utente finale di percepire quale sarà il risultato finale della realizzazione.

8 Disegno di dettaglio

Sono previste due tipologie del documento Disegno di dettaglio:

- Disegno di dettaglio di applicazione;
- Disegno di dettaglio di obiettivo.

I documenti differiscono per l'ambito di riferimento: il primo l'intera applicazione, il secondo l'obiettivo. Entrambi i documenti formalizzano le funzionalità suddividendole e organizzandole in moduli elaborativi strutturati. Deve essere inclusa nel disegno di dettaglio la formalizzazione dello schema logico e fisico dei dati e il modello dinamico delle entità secondo gli standard adottati da CSEA. Alcune delle informazioni che, per i vari moduli, devono essere presenti sono le seguenti:

- descrizione delle funzioni svolte;
- tipologia (on-line, batch, etc..);
- indicazioni sulla riutilizzabilità del componente;
- parametri scambiati con altri componenti;



Allegato Tecnico 2	
Standard tecnologici e architettonici	
Evoluzione del Sistema Informativo della CSEA	

- parametri di attivazione;
- accessi agli archivi/base dati;
- controlli e diagnostica;
- algoritmi di calcolo.

In ogni caso dovranno essere prodotte le matrici d'uso (o matrici CRUD) degli archivi da parte dei moduli software. Nei casi critici, per dimensioni delle basi dati e/o frequenza di utilizzo, deve essere indicata la frequenza prevista per il tipo d'uso che il modulo fa degli archivi/basi dati, le frequenze totali per tipo d'uso relative a ciascun archivio/tabella della base dati, le frequenze totali per tipo d'uso per ciascun modulo. Per quanto riguarda il caricamento iniziale dei dati, dovranno essere indicati:

- gli archivi fisici/basi dati da dove prendere i dati e il loro tracciato;
- i tracciati dei dati da caricare manualmente;
- le relazioni tra archivi fisici/basi dati e schemi logici;
- i volumi trattati, con dettaglio sulla occupazione di memoria e spazio disco;
- le modalità di inizializzazione degli archivi/basi dati.

Deve comunque essere garantita la tracciabilità con il documento di Specifica dei requisiti ed il glossario. I dati contenuti nel documento devono essere sempre tenuti aggiornati.

9 Convalida della tecnologia

Il documento attesta la conformità di quanto analizzato / progettato alle indicazioni del produttore della tecnologia/prodotto stesso. Esso dovrà essere prodotto per gli obiettivi che fanno uso di specifiche ed individuate tecnologie/prodotti (come riportati nel Piano della qualità generale o di obiettivo). Tale documento dovrà esplicitare:

- il nome e la release dei prodotti utilizzati;
- i puntuali riferimenti (manualistica, best practices, indicazioni specifiche, ecc.) su cui si baserà la realizzazione;
- la dichiarazione del Fornitore di utilizzare i prodotti secondo le specifiche valide per le versioni indicate.

10 Documentazione Dati

La documentazione dati contiene la descrizione e la rappresentazione della base dati ed esplicita eventuali collegamenti le regole tecniche con cui l'applicazione scambia flussi informativi con altre applicazioni. La documentazione dati è obbligatoriamente articolata nelle seguenti componenti:



Allegato Tecnico 2	
Standard tecnologici e architettonici	
Evoluzione del Sistema Informativo della CSEA	

- Schema concettuale;
- Schema logico;
- Mapping concettuale-logico.

11 Modello dei Dati

Il modello dei dati è composto da:

- glossario, che dovrà contenere:
 - descrizione di tutti gli oggetti degli schemi concettuali;
 - descrizione di tutti gli oggetti degli schemi logici;
 - mapping schema concettuale- logico.
- schema concettuale e logico;
- mapping concettuale-logico;
- schema fisico. Lo schema concettuale dovrà contenere le seguenti informazioni:
 - schema grafico rappresentante le entità e l'associazione tra esse intercorrenti;
 - nome (e/o codice) e descrizione del significato delle entità;
 - nome (e/o codice) e descrizione del significato delle associazioni intercorrenti tra le entità;
 - nome (e/o codice) e descrizione del significato degli attributi appartenenti alle singole entità e associazioni.

Lo schema logico dovrà contenere:

- schema grafico rappresentante le relazioni;
- vincoli di integrità;
- relazioni fondamentali;
- relazioni associative;
- chiavi primarie e secondarie.

Il mapping concettuale-logico dovrà contenere la corrispondenza tra le entità e associazioni descritte nello schema concettuale e le relazioni descritte nello schema logico.

Lo schema fisico dovrà contenere:

- indicazione del metodo di accesso utilizzato e dell'organizzazione dei dati;



Allegato Tecnico 2	
Standard tecnologici e architetture	
Evoluzione del Sistema Informativo della CSEA	

- bloccaggio di ciascun data-set;
- clausole di storage;
- descrizione dei dati interni del DBMS (tabelle, indici, ecc.) che realizzano la struttura prevista.

11.1 Dizionario Dati

Il dizionario dati dovrà contenere:

- nome della tabella;
- nome dell'attributo;
- indicazione della chiave primaria;
- tipo e dimensione dell'attributo (char, number, date ecc.);
- descrizione dell'attributo;
- dominio;
- nel caso di campi calcolati l'algoritmo di valorizzazione;
- descrizione dei codici di errore di tutti i controlli.

12 Codice sorgente

Per codice sorgente si intende a titolo esemplificativo e non esaustivo:

- programmi;
- tracciati e definizioni dati;
- schemi di input/output;
- pagine web;
- procedure;
- job;
- query;
- script (anche gli script relativi ai test automatizzati);
- utility di modifica/aggiornamento dati.

Fanno parte del codice sorgente le procedure di consegna e trasferimento oggetti per gli ambienti di configuration management, nonché le procedure di creazione delle tabelle ed i relativi job di caricamento dati (per intero DB e/o porzioni secondo criteri definiti) anche per gli ambienti di sviluppo, manutenzione, collaudo ed esercizio. Fanno parte del codice sorgente, l'help on-line e l'eventuale codice di test e collaudo.



Allegato Tecnico 2	
Standard tecnologici e architettonici	
Evoluzione del Sistema Informativo della CSEA	

In generale, il codice sorgente dovrà includere anche il codice per la distribuzione automatizzata, comprensivo di:

- procedura di installazione (setup applicazione e/o patch);
- procedura di disinstallazione;
- parametri di configurazione dell'ambiente su cui l'applicazione sarà installata.

Il codice sorgente di nuova realizzazione (anche nuovo codice all'interno di programmi preesistenti) dovrà essere redatto in conformità agli standard previsti e comunque sempre secondo le indicazioni presenti nella documentazione ufficiale dei linguaggi utilizzati.

13 Documentazione utente

La documentazione utente, rivolta all'utente finale delle applicazioni, è composta dal Manuale utente (rilasciato con il codice sorgente).

Il Manuale utente deve fornire una descrizione generale dell'applicazione e una guida operativa all'utilizzo delle singole funzionalità utilizzabili. La descrizione deve contemplare:

- la tipologia di utenza cui è destinata e le funzioni abilitate a ciascuna tipologia;
- gli eventuali flussi di dati scambiati con altri sistemi informativi o con le specifiche tipologie di utenze;
- la descrizione delle funzioni e della navigazione tra di esse;
- la spiegazione dettagliata dell'uso delle singole funzioni di interfaccia utente (comprensiva della funzione di richiamo dell'help);
- la descrizione dei contenuti degli output della applicazione (es. stampe). La descrizione delle funzionalità disponibili deve essere completa dell'elenco di tutti i codici d'errore previsti, della messaggistica ad essi associata e delle azioni da intraprendere a fronte di ciascuna segnalazione.

14 Documento di Collaudo

Il verbale di accettazione viene redatto dal Fornitore nella fase di Collaudo; è sottoscritto dal Fornitore e da CSEA. Il verbale certifica l'esito positivo della attività di Collaudo e consente la presa in carico e il successivo rilascio in esercizio.

Tale documento dovrà indicare almeno

- gli oggetti applicativi con le versioni;
- la data di Collaudo;



Allegato Tecnico 2	
Standard tecnologici e architetture	
Evoluzione del Sistema Informativo della CSEA	

- la data di Verifica;
- responsabili;
- referenti funzionali e tecnici;
- eventuali osservazioni.

15 Manuale di gestione esercizio

Il Manuale di gestione esercizio (applicativo e sistemistico) è lo strumento necessario al personale tecnico delle strutture preposte all'installazione ed esercizio dell'applicazione. Il Manuale è rivolto a personale tecnico e dovrà essere corredato di uno schema riepilogativo contenente informazioni anagrafiche relative all'applicazione, la tipologia del DB, la dipendenza con altre applicazioni, modelli di interfaccia, i tool utilizzati per lo sviluppo. Per quello che riguarda gli ambienti di collaudo ed esercizio, il documento dovrà esplicitare i parametri di personalizzazione dei prodotti, le modalità di attuazione dei livelli di protezione dei dati, le modalità di accesso al sistema e alle transazioni, le soluzioni tecniche necessarie alla realizzazione di tali modalità. Il manuale, ove previste deve contenere le procedure off line (batch, job, stored procedure, script ecc.) quale supporto alle attività ordinarie di gestione. Si articola nei componenti di seguito riportati e dovrà:

- essere corredato da uno schema riepilogativo contenente informazioni anagrafiche relative all'applicazione, tra le quali la tipologia del DB, la dipendenza con altre applicazioni, i modelli di interfaccia, i tools utilizzati per lo sviluppo;
- esplicitare i parametri di personalizzazione dei prodotti, le modalità di attuazione dei livelli di protezione dei dati, le modalità di accesso al sistema e alle transazioni, le soluzioni tecniche necessarie alla realizzazione di tali modalità.
- contenere, ove previsto, le procedure che non rispondono a logiche "online", quali batch, job, stored procedure, script ecc.;
- contenere, ove opportuno, requisiti tecnici quali consumo di risorse elaborative, politiche di backup, grandezze e soglie di monitoraggio tecnico ed applicativo, procedure di startup e shut-down, etc..

16 CMDB

Il CMDB deve contenere un elenco di tutti gli oggetti software realizzati, modificati o resi obsoleti nell'ambito delle attività riguardanti.

Le informazioni da fornire sono:

- Applicativo;



Allegato Tecnico 2	
Standard tecnologici e architettonici	
Evoluzione del Sistema Informativo della CSEA	

- Descrizione dell'applicativo;
- Data di inserimento;
- Riferimento change management system;
- Riferimenti Collaudo;
- Riferimenti Esercizio;
- Eventuali note.

17 Capacity planning

Il "Capacity Plan" descrive ed indica i requisiti e le previsioni prestazionali e funzionali delle applicazioni, motiva e dettaglia le esigenze di evoluzione della infrastruttura ICT del sistema finalizzate al soddisfacimento dei requisiti, sulla base dell'analisi dei dati rilevati a fronte della sistematica misurazione delle prestazioni e dei volumi di carico del sistema e sulla base del volume atteso. Le analisi di andamento del carico, dell'uso delle risorse del sistema e della qualità del servizio, presenti nel report di Capacity Plan, saranno utilizzate come base per l'applicazione delle procedure di gestione dei fabbisogni di evoluzione infrastrutturale. Sulla base delle risultanze presenti nel documento di Capacity Plan dovranno essere svolte, a titolo esemplificativo e non esaustivo, le attività di seguito riportate:

- supporto alle attività di "tuning":
 - tuning del sistema operative;
 - tuning applicazioni;
 - riorganizzazione della basi dati;
 - analisi del I/O;
 - analisi del carico di lavoro online;
 - analisi del carico di lavoro batch.
- valutazione dei risultati delle suddette attività;
- individuazione piani di intervento in caso di situazioni critiche;
- valutazioni opportunamente argomentate e dimostrate del carico sui componenti hardware e software in uso.



Allegato Tecnico 2	
Standard tecnologici e architeturali	
Evoluzione del Sistema Informativo della CSEA	

18 Piano di Test

Il Piano di Test è un documento che accompagna ogni progetto lungo tutto il ciclo di sviluppo, ed è pertanto un documento che si evolve nel tempo. Ha lo scopo di definire test specifici, tramite i quali saranno sottoposti a verifica i prodotti della realizzazione, con particolare riguardo alla loro validazione rispetto ai requisiti dell'utente, nonché documentare il loro esito.

Il Piano di Test deve essere fornito in prima versione nella fase di analisi ("Piano di Test - Analisi"), per poi essere implementato ed arricchito durante le fasi di disegno ("Piano di Test - Disegno") e di realizzazione ("Piano di Test - Realizzazione").

Il Piano di Test - Realizzazione sarà di diretto supporto per il collaudo o per la preparazione di un piano specifico di collaudo.

Tale versione del Piano è input al sistema di test e collaudo.

Il "Piano di Collaudo" sarà di diretto supporto alla fase di Collaudo. Ha lo scopo di definire test specifici volti a verificare i prodotti della realizzazione, con particolare riguardo alla loro validazione rispetto ai requisiti dell'utente, nonché documentare il loro esito. Deve essere garantita la coerenza con il documento di Specifiche funzionali, Specifiche requisiti e Disegno di dettaglio.

Nella redazione del documento il Fornitore dovrà descrivere:

- le funzionalità da sottoporre a test e le condizioni generali di test, che permetteranno la verifica della coerenza delle funzioni rispetto ai requisiti espressi e inespressi;
- per ogni condizione i possibili casi di test da eseguire, le azioni da compiere per eseguire il test e i risultati attesi;
- la verifica dell'esito del test che dovrà essere ripetuto fino all'ottenimento del risultato positivo.
- Sistemi hardware e software di riferimento
- Tipo di test

Si ricorda che i test dovranno essere organizzati secondo i seguenti riferimenti:

Categorie di test:

- test unitari – unit test
- test di integrazione – integration test
- test di sistema – system test



Allegato Tecnico 2	
Standard tecnologici e architetture	
Evoluzione del Sistema Informativo della CSEA	

Test di accettazione:

- test di accettazione – acceptance test

Aspetti da testare:

- test funzionali e non funzionali
- test di usabilità
- test prestazionale – performance test
- stress test
- test di installazione

Modalità di test:

- test a black box
- test white box
- test di regressione – regression test

18.1 Categorie di Test

18.1.1 Test unitari

Il test di unità verifica il funzionamento isolato delle parti software che sono testabili separatamente. Tipicamente, il test di unità viene effettuato con accesso al codice testato e con il supporto di strumenti per il debugging ed è focalizzato su unità isolate del prodotto (moduli, classi, funzioni).

18.1.2 Test di integrazione

Il test di integrazione è il processo di verifica dell'interazione tra componenti software. Le strategie classiche di test di integrazione, top-down o bottom-up, vengono utilizzate con il software tradizionale, strutturato in modo gerarchico. Le strategie moderne di integrazione sistematica sono invece guidate dall'architettura, cioè i componenti e sottosistemi software vengono integrati sulla base delle funzionalità identificate. I test di integrazione verificano come due (o più) unità lavorano insieme.

18.1.3 Test di sistema

Il test a livello di sistema si preoccupa del comportamento di un sistema nel suo complesso. La maggior parte degli errori dovrebbe essere già stato identificato durante il test unitario e di integrazione. Il test di sistema viene di solito considerato appropriato per verificare il sistema anche rispetto ai requisiti non funzionali, come quelli di sicurezza, velocità, accuratezza ed affidabilità. A questo livello vengono anche testate le interfacce esterne nei confronti di altre applicazioni, componenti standard, dispositivi hardware e ambiente operativo.



Allegato Tecnico 2	
Standard tecnologici e architeturali	
Evoluzione del Sistema Informativo della CSEA	

18.1.4 Test di accettazione

Il testing di accettazione controlla il comportamento del sistema rispetto ai requisiti (comunque espressi) da CSEA. I clienti effettuano, o specificano, attività tipiche di uso del sistema, per controllare che i loro requisiti siano stati soddisfatti. Questa attività di test può coinvolgere o meno gli sviluppatori del sistema. Questo test è condotto in un ambiente operativo reale per determinare se un sistema soddisfa i propri criteri di accettazione (ad esempio i requisiti iniziali, e le attuali esigenze dei suoi utenti) e per permettere al cliente di determinare se accettare o meno il sistema.

18.2 Aspetti da testare

18.2.1 Test funzionali e non funzionali

Possono essere progettati dei casi di test per verificare che le specifiche funzionali siano state implementate correttamente. Queste tipologie di test sono chiamate test di conformità, o test di correttezza, o test funzionale. In ogni caso, possono essere testate anche numerose altre proprietà non funzionali, tra cui le prestazioni, l'affidabilità e l'usabilità. Il test funzionale è un tipo di test comportamentale (cioè focalizzato sul comportamento osservabile del prodotto) che guarda al sistema come ad una collezione di funzioni.

18.2.2 Test di usabilità

Il test di usabilità valuta la facilità di uso e di apprendimento del software da parte degli utilizzatori finali. Vengono presi in considerazione la documentazione per l'utente, l'efficacia di funzionamento del software nel supportare le attività dell'utente, le possibilità che il software offre all'utente di porre rimedio ai propri errori.

18.2.3 Test prestazionale

Il test prestazionale è specificamente effettuato per verificare che il software soddisfi i requisiti di prestazioni specificati, ad esempio i livelli di capacità di utilizzo ed i tempi di risposta.

18.2.4 Test di stress

Lo stress test sollecita il software al livello massimo di carico progettato, ed oltre.

18.2.5 Test di installazione

Il test di installazione consiste nella verifica dell'installazione del software nell'ambiente operativo di destinazione (effettuata di solito dopo la conclusione dei



Allegato Tecnico 2	
Standard tecnologici e architetture	
Evoluzione del Sistema Informativo della CSEA	

test di accettazione e può essere visto come un test di sistema condotto in conformità con i requisiti di configurazione hardware).

18.3 Modalità di test

18.3.1 Test black box

Il test è a scatola nera se ci si basa solo sul comportamento riscontrabile analizzando dati di ingresso e di uscita, senza conoscere in alcun modo il codice.

18.3.2 Test white box

Il test è a scatola bianca (white-box) se si basa su informazioni relative a come il software è stato progettato o codificato.

18.3.3 Test di regressione

Il test di regressione consiste nella riesecuzione selettiva di test su un sistema o un componente modificato, per verificare che le modifiche non abbiano provocato effetti indesiderati. In pratica, l'idea è di dimostrare che il software che aveva passato i test prima delle modifiche continua a farlo anche dopo.



Allegato Tecnico 2	
Standard tecnologici e architettonici	
Evoluzione del Sistema Informativo della CSEA	

19 Riepilogo della documentazione tecnica prevista tra i delivery di progetto

A fronte di ogni specifica fase di realizzazione del progetto, dovrà essere concordata con CSEA e rilasciata la documentazione specifica che può constare dei seguenti documenti:

- Modello documentale che sintetizza il progetto, il landscape, i rischi, i costi e la strategia di implementazione.
- La WBS di progetto
- Il piano delle risorse
- Il piano dei rischi
- Documentazione che definisca gli obiettivi funzionali e tecnologici di progetto, assegnazione delle responsabilità di ogni gruppo di lavoro, definizione stakeholder all'interno della struttura CSEA.
- Documentazione di analisi dei processi di business e della loro organizzazione.
- Documentazione dei requisiti di dettaglio individuati, specifiche tecniche di dettaglio
- Documentazione tecnica che riporti i dettagli di configurazione dell'applicativo sviluppato; la documentazione tecnica dovrà coprire inoltre ogni dettaglio riguardo lo sviluppo e il test di personalizzazioni non coperte dalla soluzione standard.
- Documentazione di test funzionale e di integrazione del sistema sviluppato con tutti i sistemi CSEA coinvolti dalla sua operatività.
- Documentazione tecnica e operativa pertinente alla strategia di migrazione di dati.
- Documentazione relativa all'avviamento della piattaforma integrata con il sistema CSEA, le strategie di monitoring e gestione della piattaforma.
- Il report di stato avanzamento lavori
- I report dei casi di test e dei test di accettazione
- Documentazione della gestione di eventuali interventi correttivi, la gestione delle modifiche ed il configuration management.



Allegato Tecnico 2	
Standard tecnologici e architetturali	
Evoluzione del Sistema Informativo della CSEA	

20 Ambiente di Sviluppo

20.1 Riferimenti alle versioni

Prima di iniziare le attività di sviluppo occorre predisporre l'ambiente di lavoro. Questo consiste almeno dei seguenti componenti che devono essere installati e configurati:

IDE:

- Eclipse 4.7 e superior, STS (allineato alle versioni di Eclipse)

Eclipse Plugin previsti:

- Maven Integration
- Subclipse

Software di supporto:

- JDK 1.7 e successive
- Apache Maven 3
- KETTLE
- Findbugs
- SOAPUI (Test Web Services)
- MYSQL workbench
- Tomcat 7 e successive
- Probe (monitoring tomcat)

Altro:

OS name: "Windows 7 e successive" / "Windows Server 2008 R2" / 64bit

Java version: 1.7.X e superior (Oracle Corporation)

Default locale: it_IT, **platform encoding:** UTF8

Note:

- Web Container (es: Tomcat, JBoss) e DBMS (es: MySql)
- Configuration Management System: (Subversion, GIT). Nel sistema di configuration management andrà configurato il "repository" che conterrà tutte le risorse del progetto (sorgenti, documenti, immagini, file di configurazione, ...).
- Performance Test Tool (ad es. JMeter)
- Test Management tool
- SOAP UI come piattaforma per il web service testing
- Tecnologie in fase di valutazione: Docket, GIT



Allegato Tecnico 2	
Standard tecnologici e architeturali	
Evoluzione del Sistema Informativo della CSEA	

21 Documentazione

La documentazione tecnica che deve essere prodotta a fronte dello sviluppo di un Progetto deve sempre essere approvata e verificata da ASI.

Tutti i documenti dovranno essere particolarmente curati negli aspetti di:

- comprensibilità;
- accuratezza;
- adeguatezza;
- aderenza.

Si richiede particolare attenzione al versioning della documentazione. La causa di innalzamento della versione, con esplicito riferimento all'evento che lo richiede (esigenze utente o altro) deve essere sempre verificabile.

Il documento Disegno architeturale deve contenere la documentazione (secondo schemi UML/OMT) sia delle soluzioni architeturali in essere sia come output all'attività a supporto dei progetti di sviluppo, contestualizzato nell'architettura dell'ambiente di produzione.

21.1 Javadoc

Di seguito si descrivere uno standard di codifica e documentazione per il linguaggio JAVA conforme allo standard presentato dalla SUN su "The Java Language Specification" (<https://docs.oracle.com/javase/specs/>).

Javadoc è un tool che processa i sorgenti java e produce documentazione esterna in formato HTML. Il contenuto della documentazione prodotta è ottenuto dal tool estrapolando dal sorgente tutti i commenti che rispettano il seguente formato:

```
/**  
<documentazione>  
<nome tag> <valore tag>  
*/
```

I tag possono essere usati per inserire particolari informazioni inerenti l'oggetto che si sta descrivendo come l'autore o i parametri passati ad un metodo. La seguente tabella mostra i principali tag, il contesto in cui vengono usati e la loro funzione.

Tag	Contesto	Funzione
@author <nome>	Classi, Interfacce	Indica l'autore
@exception <nome> <description>	Metodi	Descrive le eccezioni lanciate dal metodo
@param <nome> <descrizione>	Metodi	Descrive i parametric passati al metodo
@return <descrizione>	Metodi	Descrive il valore ritornato dal metodo



Allegato Tecnico 2	
Standard tecnologici e architetturali	
Evoluzione del Sistema Informativo della CSEA	

@see <nome classe>	Classi, Metodi, Interfacce, Attributi	Genera un link alla documentazione della classe specificata
@see <nome classe>#<nome metodo>	Classi, Metodi, Interfacce, Attributi	Genera un link alla documentazione per il metodo specificato
@version <versione>	Classi, Interfacce	Descrive la versione

21.2 Convenzioni sulla documentazione

Header

Ogni metodo deve avere una intestazione (Header) che contiene le seguenti informazioni:

1. Cosa fa il metodo.
2. Quali sono i parametri in input. (tag @param di javadoc)
3. Cosa ritorna il metodo. (tag @return di javadoc)
4. Le eccezioni che lancia il metodo. (tag @exception di javadoc)
5. Eventuali decisioni critiche circa la visibilità del metodo.
6. Pre e Post condizioni.

Documentazione interna

All'interno dei metodi dovrebbero essere documentate:

1. Le strutture di controllo
Una o due linee di commento prima di ogni condizione o ciclo.
2. Le parti di codice critiche da comprendere.
3. Le variabili locali.
Un commento dopo la dichiarazione di ogni variabile locale.
(es: int timeStamp; // Time Stamp di accesso in scrittura).



Allegato Tecnico 2	
Standard tecnologici e architettonici	
Evoluzione del Sistema Informativo della CSEA	

22 Codifica

Il codice contenuto in un blocco (tutto ciò che è presente tra {}) dovrebbe essere indentato di una unità (tab o spazio) rispetto alle parentesi.

- Un comando per linea. Si dovrebbe scrivere solo un comando/dichiarazione per linea
- Uso di linee vuote.

Per aumentare la leggibilità del codice dovrebbero essere inserite:

- una linea vuota tra due blocchi di codice logicamente distinti
- due linee vuote tra una dichiarazione di un metodo e un'altra

22.1 Standard per i metodi

22.1.1 Convenzioni sui nomi (ad esclusione dei costruttori)

1. Il nome dei metodi dovrebbe essere più esplicativo possibile.
2. Si possono utilizzare più parole concatenate. La prima parola, completamente in minuscolo, è in genere un verbo attivo. Le successive, con l'iniziale in maiuscolo, sono in genere sostantivi.
(es: stampaFattura(); chiudiPianoDiGestione();)
3. I metodi di accesso agli attributi in genere iniziano con get e set
(es: getLineaDiAttivita(); setPianoDiGestione();)
4. I metodi che restituiscono valori booleani dovrebbero iniziare con is, has, can, exists .
(es: isPianoDiGestioneChiuso();)

22.1.2 Visibilità

Per diminuire l'accoppiamento tra classi una buona regola generale è essere più restrittivi possibile sulla visibilità dei metodi e quindi di ridurre al minimo il numero dei metodi pubblici.

22.2 Standard per gli attributi

22.2.1 Convenzione sui nomi

Si possono utilizzare più parole concatenate. La prima parola completamente in minuscolo, le successive con l'iniziale in maiuscolo. Per quanto riguarda le strutture dati come liste o collezioni vengono usati nomi al plurale (es: clienti, terzi, ecc.).

22.2.2 Visibilità

La visibilità sugli attributi dovrebbe essere più restrittiva possibile.

Gli attributi dovrebbero essere dichiarati private e accessibili solo tramite i metodi di accesso.



Allegato Tecnico 2	
Standard tecnologici e architeturali	
Evoluzione del Sistema Informativo della CSEA	

22.2.3 Convenzioni sulla documentazione

1. Un commento dopo la dichiarazione di ogni attributo. (es: `int timeStamp; // Time Stamp di accesso in scrittura`).
2. Eventuali decisioni critiche circa la visibilità dell'attributo.

22.3 Standard per i parametri

22.3.1 Convenzione sui nomi

La convenzione sui nomi è identica a quella per i parametri.

22.3.2 Convenzioni sulla documentazione

1. Specificare per cosa è utilizzato il parametro all'interno del metodo.
2. Specificare il dominio dei valori ammissibili per il parametro (solo valori positivi, solo stringhe di lunghezza inferiore a 5 caratteri).

22.4 Standard per le variabili locali

22.4.1 Convenzione sui nomi

La convenzione sui nomi è identica a quella per i parametri.

22.4.2 Convenzioni sulla dichiarazione variabili e sulla documentazione

Di seguito sono indicate le convenzioni sulla documentazione

1. Dichiarare una sola variabile per linea di codice.
2. Documentare ogni variabile con un commento a fine linea.
3. Dichiarare ogni variabile subito prima del suo uso.
4. Usare le variabili locali per una cosa alla volta.

22.5 Standard per le classi

22.5.1 Convenzione sui nomi

Si possono utilizzare più parole concatenate. L'iniziale di ogni parola dovrebbe essere in maiuscolo. Per il resto della parola non ci sono regole.

22.5.2 Convenzione sulla documentazione

Le seguenti informazioni dovrebbero apparire nella documentazione che precede la dichiarazione della classe:

1. Specificare le funzionalità generali della classe e se la classe fa parte di un pattern



Allegato Tecnico 2

Standard tecnologici e architettonici

Evoluzione del Sistema Informativo della CSEA

2. Specificare la storia dello sviluppo/manutenzione della classe in una tabella che mostra per ogni attività di manutenzione correttiva, adeguativa, migliorativa una descrizione sintetica, gli autori e la data dell'attività.

22.5.3 Convenzione sulla dichiarazione

Le classi dovrebbero essere dichiarate rispettando sempre lo stesso ordine per la dichiarazione dei metodi e degli attributi. Non importa l'ordine che viene scelto l'importante è che venga rispettato all'interno di tutto il progetto.

Una possibilità potrebbe essere:

- attributi privati
- costruttori
- metodi pubblici
- metodi protetti
- metodi privati.

22.6 Standard per le interfacce

22.6.1 Convenzione sui nomi

Il nome di un'interfaccia dovrebbe avere il prefisso I seguito da un nome significativo che segue le stesse regole della convenzione sui nomi per le classi.

22.6.2 Convenzione sulla documentazione

Le seguenti informazioni dovrebbero apparire nella documentazione che precede la dichiarazione dell'interfaccia:

L'obiettivo dell'interfaccia e la logica che incapsula. Una descrizione generica delle modalità di utilizzo dell'interfaccia. (Tale descrizione sarà poi particolareggiata per i singoli metodi usando la convenzione sulla documentazione dei metodi descritta sopra)

22.7 Standard per i packages

22.7.1 Convenzione sui nomi

1. I qualificatori vengono separati da punti.
2. I packages locali usano qualificatori in minuscolo
3. I packages che devono essere distribuiti iniziano con il dominio della società/ente che li sviluppa al contrario (es: it.csea.*).

22.7.2 Convenzione sulla documentazione

Ogni package dovrebbe avere della documentazione esterna che riguarda:

1. Principali funzionalità contenute nel package
2. Lista delle classi contenute nel package



Allegato Tecnico 2	
Standard tecnologici e architettonici	
Evoluzione del Sistema Informativo della CSEA	

22.7.3 Organizzazione del codice sorgente

É necessario definire inizialmente un'alberatura di folder e sub-folder che venga condivisa e successivamente rispettata dal team per tutta la durata del progetto. Un esempio può essere il seguente.

src\it\ccse\\dao (contenente i dao)

src\it\ccse\\exceptions (contenente le eccezioni)

src\it\ccse\\dto (contenente i dto)

src\it\ccse\\utils (contenente le classi di utilità comune)



Allegato Tecnico 2	
Standard tecnologici e architetturali	
Evoluzione del Sistema Informativo della CSEA	

23 Prospetti del modello applicativo e processo di sviluppo

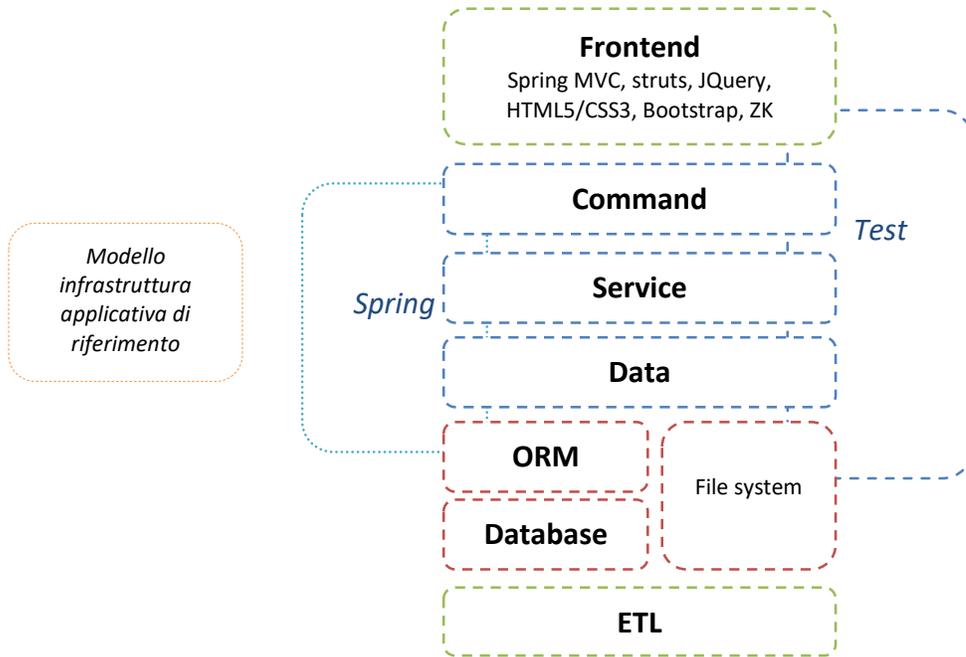


Figura 1 - Modello dell'infrastruttura applicativa



Allegato Tecnico 2	
Standard tecnologici e architetturali	
Evoluzione del Sistema Informativo della CSEA	

Processo di sviluppo software

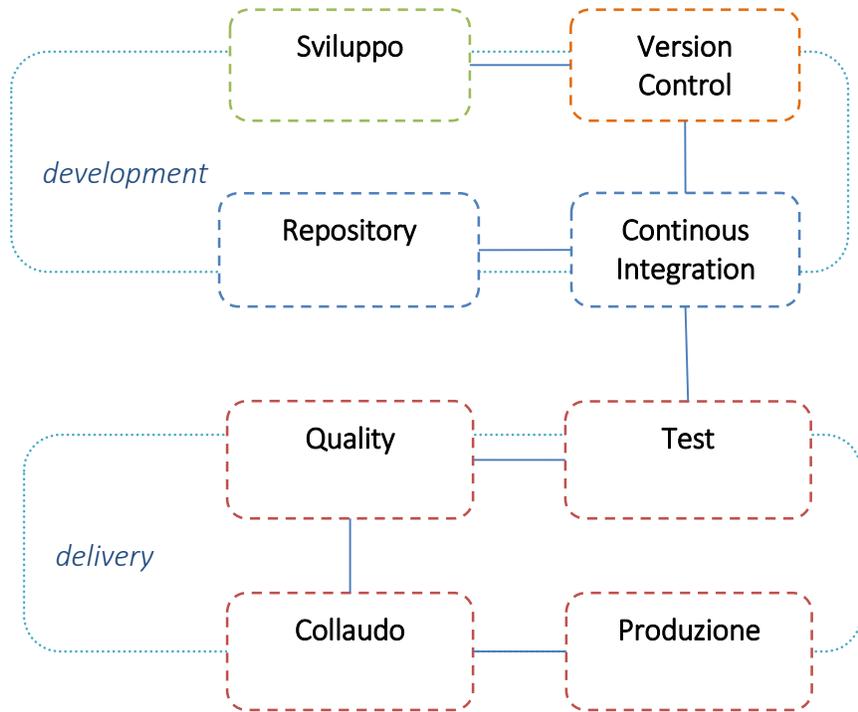


Figura 2 - Processo di sviluppo software