REGIONE MARCHE

Standard di riferimento per lo sviluppo software

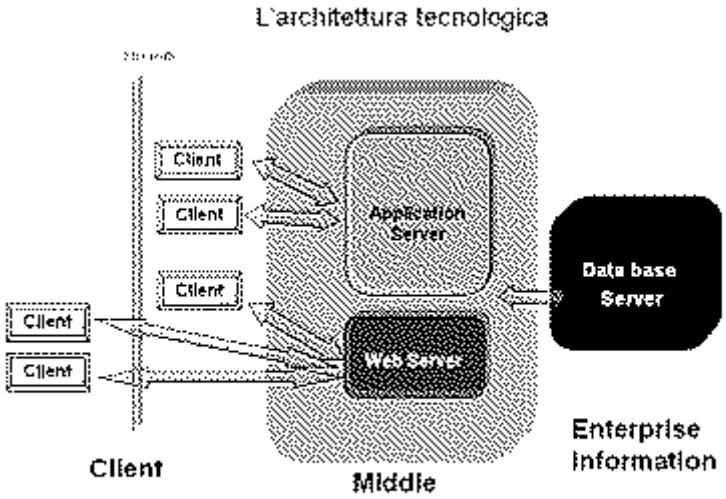
Sommario

1. ARCHITETTURA SOFTWARE DI RIFERIMENTO

Il sistema software deve avere un'architettura tecnologica "full internet/web based" ovvero un'architettura omogenea, distribuita e scalabile su tre livelli elaborativi distinti secondo il modello "thin-client" e "web-based":

- livello "**Presentazione**" con interfaccia utente grafica evoluta GUI operante su piattaforma client che richieda la presenza del solo componente web-browser;
- livello "**Applicazione**" dove è operativa la logica applicativa del sistema, posizionato su una macchina logica denominata appunto "application server";
- livello "**Dati**" dove opera il gestore di database posizionato su un'altra macchina logica denominata in questo caso "database server".

Il modello richiesto è schematizzato dalla figura seguente:



MODELLO ARCHITETTURA SOFTWARE A TRE LIVELLI

Tutte le componenti software dovranno essere completamente e nativamente integrate e supportate dello stesso stack tecnologico.
Le componenti applicative dovranno essere percepite dai vari utilizzatori come un sistema unico: l'interfaccia utente e la logica di funzionamento del sistema dovranno essere quindi comuni all'intera soluzione applicativa che quindi dovrà presentare maschere, modalità operative, parametri, tasti funzione ecc. tra loro congruenti e consistenti, indipendentemente dalle funzionalità associate.
Il protocollo di comunicazione tra i livelli, su rete geografica e locale, dovrà necessariamente essere il TCP/IP aderenti agli standard "internet" e compatibili con l'interposizione di firewall e la realizzazione quindi di DMZ sicure.
Vediamo ora nel dettaglio le specifiche di ciascun livello.

1.1 Specifiche del livello "Presentazione"

Questo livello dovrà essere in grado di soddisfare le esigenze di ogni tipologia di utenza, garantendo

sempre l'accesso al sistema tramite l'uso esclusivo di web-browser standard. In particolare dovranno essere soddisfatte le esigenze di:

- tipologie di utenza "esperta" (licenze full) che utilizza quotidianamente il sistema e che quindi necessita di un'interfaccia utente efficace che sia in grado di garantire un elevato livello di transazionalita' e che riduca al minimo la navigazione tra le maschere.
- altre tipologie di utenza che utilizzano il sistema saltuariamente, eventualmente accedendo a funzioni applicative ridotte; tali utenti dovranno essere supportati da un'interfaccia utente dotata di "wizard" ed estremamente guidata.

A tutte le tipologie di utenza dovranno essere resi disponibili meccanismi d'ausilio all'interazione con il sistema (ad esempio: meccanismi di validazione campo per campo, liste di valori per facilitare l'inserimento dei dati, menù dinamici e finestre multiple coordinate) al fine di prevenire errori, risparmiare tempo e ridurre necessità di training; inoltre tutte le tipologie di interfaccia utente, indipendentemente dal modulo applicativo a cui appartengono, dovranno insistere sui medesimi dati fisici gestiti dal medesimo database all'interno dello stesso modello dati logico.

1.2 Specifiche del livello "Applicazione"

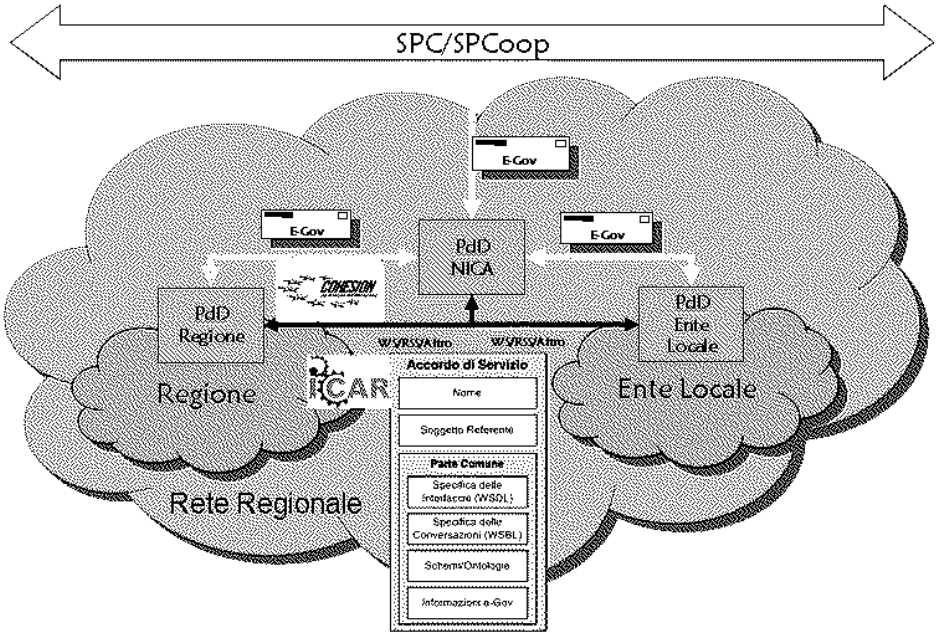
Il livello "Applicazione", dovrà esclusivamente governare la logica applicativa legata alla presentazione dei dati. In tal senso, da tale livello, non dovranno essere gestite repliche locali di dati ma dovranno essere acceduti e modificati solo e direttamente i dati residenti nel database gestito dal livello "Database".
Le tecnologie di sviluppo della logica applicativa e dell'interfaccia utente legate al livello "Applicazione" dovranno essere caratterizzate da ampia diffusione nella comunità ICT.

1.3 Specifiche del livello "Database"

Il Database deve essere necessariamente di tipo relazionale.
Deve essere garantita l'univocità fisica dei dati comuni a più moduli applicativi e la congruenza dei dati.

2. ARCHITETTURA DI COOPERAZIONE APPLICATIVA

Il modello di cooperazione applicativa si basa sul progetto ICAR che prevede l'utilizzo delle porte di dominio all'interno della rete regionale ed il modello di autenticazione basato su SSO Cohesion secondo lo schema seguente:



Tale sezione è stata esplicitata *nell'allegato "Infrastrutture tecnologiche abilitanti della Regione Marche".*

3 - MODALITÀ DI SVILUPPO DELLE FUNZIONALITÀ SOFTWARE.

Per tutte le attività di sviluppo o modifica del software dovrà essere utilizzata una metodologia di sviluppo che preveda almeno le 3 fasi iterative-incrementali che seguono, ferme restando le migliorie previste dall'offerta tecnica:

1. **Fase di Inception ed Elaboration**: La realizzazione delle nuove funzionalità deve rispettare in linea di massima la priorità assegnata nel Progetto esecutivo e comunque deve essere concordata con il Direttore dell'esecuzione. In corrispondenza della medesima priorità, il fornitore deve individuare un buon insieme di requisiti (80% dei complessivi) e definire l'architettura di base. Alla fine di questa fase si avrà una stima dettagliata dei tempi necessari per lo sviluppo e quindi dell'effort necessario.
2. **Fase di Construction**: sviluppo del prodotto secondo una o più versioni "beta".
3. **Fase di Transition**: rilascio del prodotto pronto per la fase di verifica di conformità e quindi per il deployment successivo.

Per ciascuna iterazione pianificata nel progetto esecutivo si segue il flusso tipico dello sviluppo software:

Analisi → Progettazione → Codifica → Test

Il passo di analisi dovrà produrre un documento secondo il formato dettagliato nel paragrafo successivo.
Il test deve essere effettuato sulle nuove funzionalità dell'iterazione e, soprattutto, su quelle precedentemente realizzate.
Il prodotto di ciascuna iterazione (prototipo), dopo aver superato il test dell'Affidatario viene essere rilasciato, in ambiente di test, al Committente ovvero dall'Utente delegato dal Direttore dei lavori che provvederà a verificare la conformità a quanto richiesto.
La lista delle eventuali difformità "gravi" ovvero che pregiudicano lo sviluppo delle parti successive, dovrà essere comunicata tempestivamente al RA e comunque entro la metà del tempo che preventivato per la consegna della fase iterativa successiva in modo da consentire di svolgere le necessarie correzioni.
Nel caso che la lista delle difformità gravi venga prodotta in ritardo, L'affidatario ha la facoltà di ritardare la consegna della prossima iterazione di tanti giorni quanti sono i giorni di ritardo della comunicazione.
La lista delle difformità lievi dovranno essere comunque consegnate entro il successivo rilascio e quindi potranno essere effettuate entro il successivo rilascio incrementale ovvero prima della validazione finale della funzionalità.
La qualità complessiva del rilascio definitivo di una funzionalità sarà misurato valutando il rapporto Q= DIFETTI/NOP dove:

a) DIFETTI è la somma tra il numero di difetti gravi lamentati in ciascuna fase iterativa, quelli riscontrati nel collaudo finale e quelli (ad alta priorità) lamentati nei primi 90 giorni di esercizio;
b) NOP è la dimensione del software sviluppato misurato in Object Point al netto del riuso così come definiti nel modello Cocomo II e dettagliato nel seguito

Si richiede che sia Q < 3 altrimenti il Direttore dei lavori ha la facoltà di applicare le penali previste.

3.1 - Documento di analisi dei requisiti

Il documento di analisi dei requisiti dovrà seguire lo standard IEEE 830-1998 e classificare i requisiti funzionali e non funzionali secondo il modello denominato "FURPS+" (originalmente sviluppato da Robert Grady e Deborah Caswell) ovvero dovranno essere distinti i requisiti:

- **F**UNZIONALI - sono i requisiti derivabili direttamente dal modello dei casi d'uso ed ai quali ne possono essere aggiunti altri non espressi nei diagrammi per esigenza di sintesi.

- **U**SABILITA' - si riferiscono a standard (inter)nazionali di usabilità (es. Legge "Stanca" sull'accessibilità dei siti web), ai tempi massimi per task misurabili ed ai tempi necessari per la formazione.
- **R**ELIABILITY – si riferiscono all'affidabilità del sistema come ad esempio: disponibilità giornaliera, MTBF, MTTR, accuratezza e precisione, difetti critici e non, …
- **P**RESTAZIONI – ad esempio: tempi di risposta per determinate transizioni, numero transazioni/secondo, numero massimo di utenti, utilizzo risorse (memoria, disco, banda, ..), massimo degrado sostenibile temporaneamente, …
- **S**OSTENIBILITÀ - sono i requisiti che rendono i costi per la realizzazione e la manutenzione sostenibili economicamente: uso di standard, convenzioni, componenti open-source, metodologie di manutenzione, …
- **+** - tutti gli altri requisiti che non sono stati classificati precedentemente e rappresentano dei "vincoli" di progetto. Ad esempio: linguaggi, processi, tool di sviluppo, ecc.

Ciascun requisito dovrà essere quindi "qualificato" secondo i seguenti attribuiti:

- **S**tato: indica lo stato di approvazione del requisito da parte dei committenti. Lo stato evolve dall'iniziale assegnazione durante tutta la vita del progetto. I possibili stati significativi sono:
  - P Proposto: descrive il requisito che non è ancora stato rivisto e accettato dal canale ufficiale;
  - A Approvato: il requisito è utile e fattibile ed è stato approvato ufficialmente;
  - I Incorporato: il requisito è già esistente nella versione attuale del sistema.
- **B**enefici: il beneficio viene impostato inizialmente dall'analista ma deve essere verificato ed accettato dai canali ufficiali. Il valore viene usato per gestire la portata del progetto e per determinare le priorità di sviluppo. Si consiglia di usare la seguente scala:
  - 5 Critico: indica un requisito essenziale. Non implementarlo significa che il sistema non risolverà le necessità del committente;
  - 4-3 Importante: indica un requisito importante per la consistenza e l'efficienza del sistema per molte applicazioni. Tale funzionalità non può essere facilmente sostituita da altre strade;
  - 2-1 Utile: indica un requisito utile in alcune situazioni minori, usate meno frequentemente e può essere superata con azioni sostitutive ragionevolmente efficienti;

- **C**osto (effort): questo attributo misura il costo per implementare il requisito in termini di Object-Point al netto del riuso (NOP), Il costo serve anche per valutare il miglior rapporto costo/prestazioni ovvero la priorità nei requisiti;
- **R**ischio: questo indica la probabilità che, implementando tale requisito, il progetto vada incontro ad eventi indesiderati come costi extra, ritardi o cancellazioni. La valutazione viene spesso fatta misurando l'incertezza della stima del costo di una certa specifica. Per valutare il rischio usiamo la scala:
  - 5 Critico: indica tale requisito che deve essere necessariamente bilanciato da un altro che ne mitighi la pericolosità;
  - 4-3 Importante: indica un rilevante livello di rischio nel soddisfare tale requisito che può essere mitigato, più o meno facilmente, con interventi preventivi o con controlli puntuali.
  - 2-1 Basso: difficilmente si avranno dei problemi nel soddisfare tale requisito.
- **I**nalterabilità: indica la probabilità che tale requisito, o la sua comprensione, cambierà nel tempo; serve per stabilire le priorità e quali sono gli argomenti da approfondire. Possiamo usare la seguente scala:
  - 5 Stabile: indica un requisito chiaramente conosciuto, immutabile e senza incertezze di interpretazione;
  - 4 Normale: indica un requisito ritenuto sufficientemente stabile anche se non è stata ancora verificata l'effettiva stabilità;
  - 3-2 Problematico: è il caso in cui il requisito è soggetto ad interpretazione soggettiva e quindi occorre predisporre un'opportuna documentazione chiarificatrice o ulteriori approfondimenti;
  - 1-0 Instabile: il requisito può essere soggetto a modifiche normative o non è ancora ben definita la necessità del committente.
- **F**ase prevista (Target Release): indica in quale iterazione o scadenza temporale si prevede sarà soddisfatto tale requisito. In prima approssimazione solo i requisiti con stato "Incorporato" saranno

senz'altro assegnati alla release iniziale. L'indice indica comunque un livello di priorità attribuito al requisito, dove 1 indica la priorità massima.

- **M**otivazione: serve per tenere traccia dell'origine della specifica richiesta. Utile per evitare di dimenticare la ragione per cui si sta facendo lo sforzo per rispettare il requisito ed eventualmente cancellare i requisiti collegati a mutate condizioni o necessità. In genere si fa riferimento ad allegati o altri punti del documento di analisi.

3.2 - Modalità di stima degli "Object Point"

In base alle specifiche stabilite dal modello Cocomo, a titolo di riferimento, si riportano le modalità con cui dovranno essere stimati gli "Object Point" relativi allo sviluppo software:

1) Si stima il numero di schermate, reports e moduli elaborativi da sviluppare.
2) Si classificano gli oggetti in "semplici", "medi" e "difficili" in base alla loro dimensione. In particolare:
   a) Per le schermate, si conteggiano il numero di viste ed il numero di tabelle "server" (S) e di tabelle client (C) e quindi si classifica utilizzando lo schema seguente:

Numero di viste	**Sorgenti tabelle e dati**		
	< 4 (< 2 S, < 3 C)	<8 (2-3 S, 3-5 C)	> 8 (> 3 S > 5 C)
< 3	Semplice	Semplice	Medio
3 - 7	Semplice	Medio	Difficile
>7	Medio	Difficile	Difficile

   b) Per i reports, si conteggiano il numero di sezioni ed il numero di tabelle "server" (S) e di tabelle client (C) e quindi si classifica utilizzando lo schema seguente:

Numero di sezioni	**Sorgenti tabelle e dati**		
	< 4 (< 2 S, < 3 C)	<8 (2-3 S, 3-5 C)	> 8 (> 3S > 5 C)
0-1	Semplice	Semplice	Medio
2 - 3	Semplice	Medio	Difficile
>3	Medio	Difficile	Difficile

   c) I moduli elaborativi rappresentano l'oggetto di collegamento tra le videate ed i reports e corrispondono in genere a quanto necessario per una voce di menù di primo livello. Se sono necessarie particolari "elaborazioni" ovvero trasformazioni dati complesse, si utilizza il criterio semplificato di identificare un modulo elaborativo per ogni 500 campi elaborati/trasformati.

c) In base al numero di oggetti e della loro relativa classificazione, si conteggia il numero di Object Point moltiplicando ciascun oggetto per il peso corrispondente riportato nella seguente tabella:

OBJECT POINTS	Semplice	Media	Difficile
Schermate	1	2	3
Reports	2	5	8
Moduli elaborativi			10

E quindi sommando tutti i prodotti parziali.

d) Per quanto riguarda la stima degli eventuali progetti di Business Intelligence, si utilizzeranno seguenti criteri di classificazione:

Numero di misure	Numero di tabelle + dimensioni		
	< 30	30 .. 50	> 50
1	semplice	semplice	Medio
2 - 3	semplice	medio	Difficile
>3	medio	Difficile	Difficile

E quindi la seguente valutazione in Object Point per ogni “Cubo Olap”:

OBJECT POINTS	Semplice	Media	Difficile
Cubi Olap	1	3	5

3) Il valore complessivo ottenuto in base alle regole precedenti va infine ridotto, applicando una percentuale associata al “riuso” di oggetti già esistenti (che può andare, mediamente, dal 20% al 80%) per ottenere il valore finale (NOP = (1-riuso) x OP) da conteggiare come effort per lo sviluppo del progetto. Per ottenere il corrispondente valore di effort in gg-uomo, sarà necessario applicare un coefficiente di produttività che può variare, a seconda delle competenze e degli strumenti di sviluppo utilizzati, da 4 NOP/mese (molto bassa) a 50 NOP/mese (molto alta).

Per ulteriori dettagli sulla metodologia sopra illustrata, consultare il sito ufficiale del modello “Cocomo”:
http://sunset.usc.edu/csse/research/COCOMOII/cocomo_main.html

4 - IL SISTEMA DI CONTENT MANAGEMENT - CMS

La Regione Marche si è da tempo dotata di un sistema di content management (CMS) open source denominato dotnetnuke (.net nuke o DNN), per la gestione distribuita delle informazioni pubblicate nei siti web regionali. Tale soluzione, della quale attualmente si impiega la versione 6.x "community edition", è basata sull'architettura Microsoft .net, application/web server IIS e RDBMS Microsoft SQL Server. Risulta liberamente scaricabile dal sito www.dotnetnuke.com della DNN Corporation.
Il software dotnetnuke è utilizzato come sistema di redazione e pubblicazione presso tutti i principali portali istituzionali, tematici o di struttura recentemente realizzati, internamente o da ditte esterne che si sono aggiudicate appalti di tale natura, per conto della Giunta regionale.
La PF Sistemi Informativi e Telematici nei propri siti web implementa di norma moduli dotnetnuke in lingua italiana standard gratuiti (es. text/html, blog/news, etc.) o acquisiti sul mercato (es. newsletter), e dispone inoltre di numerosi moduli personalizzati al fine di gestire contenuti dinamici su database, raccogliere informazioni dagli utenti mediante form (es. sottoscrizione convegni) o realizzare soluzioni avanzate (es. autenticazione integrata con il sistema regionale Cohesion/SSO, features per riallineare gli skin e l'html automaticamente generato ai requisiti di accessibilità secondo la normativa Stanca).
Il CMS può essere utilizzato anche come interfaccia web / presentation layer di sistemi informativi web-based. Naturalmente la logica di accesso e scrittura dati e le altre funzionalità avanzate (non supportate da moduli standard o già disponibili) andranno implementate attraverso moduli DNN personalizzati in asp.net C#.

4.1 - Requisiti comuni

Le caratteristiche applicative ed architetturali di ogni sistema sviluppato al, dovranno rispondere pienamente alla definizione degli elementi che rendono "Riusabile" un sistema o applicazione software nella Pubblica Amministrazione.
I riferimenti di compatibilità riguardano le "caratteristiche tecniche del sw riusabile":
− Sviluppo su "layer" successivi (riuso a vari livelli)
− Modularità
− Bassa dipendenza dalla piattaforma (J2EE,…)
− Indipendenza dei cambiamenti
− Adozione di specifiche cosiddette "Reusable Asset Specification" (RAS) per lo sviluppo sw.
Per quanto riguarda la disponibilità della piattaforma del sw applicativo sviluppato, si ritiene necessario avere sempre la piena disponibilità dei sorgenti e la proprietà del software al fine di garantire la capacità di intervento immediata sull'applicativo e l'eventuale indipendenza dell'Amministrazione dai fornitori esterni, per il suo adeguamento.
In ogni caso, si potrà prevedere, a seguito di specifici accordi, che il fornitore esterno utilizzi lo stesso software in realtà differenti.

4.2 - Requisiti relativi alla sicurezza

Per gli aspetti di sicurezza, sono da assumere a riferimento tutte le specifiche e le linee guida pubblicate dal CNIPA.
Si segnalano:
- le "Linee guida provvisorie per l'applicazione dello schema nazionale per la valutazione e certificazione di sicurezza nel settore della tecnologia dell'informazione";
- le "Linee guida per l'utilizzo della Firma Digitale" e per l'utilizzo della Carta Nazionale dei Servizi (CNS);

- le specifiche tecniche pubblicate dal CNIPA.

Si segnalano inoltre:

- le “Regole tecniche e di sicurezza relative alle tecnologie e ai materiali utilizzati per la produzione della Carta Nazionale dei Servizi” (DPCM 9 dic. 2004).
- Le norme UNI EN 12251:2004.
- Linee guida dell’ISCOM.
- Pubblicazione CNIPA: Sistema pubblico di cooperazione - SERVIZI DI SICUREZZA Versione 1.

Il livello di sicurezza implementato, andrà scelto anche in base alla particolare sensibilità dei dati riguardanti soggetti fisici e giuridici.