

1- Piano di progetto di un sistema remoto per visualizzare e stampare referti:

- dare molta enfasi alla "divisione del lavoro";
- prevedere un piano di test;

specifiche del sistema...

-servizio di messaggistica per avvisare la presenza di referti

Piano di progetto

1. introduzione: descrive brevemente gli obiettivi e delinea i vincoli (budget, tempo, etc.) che influenzano la gestione del progetto;

Il progetto consiste in un sistema remoto per visualizzare e stampare referti. Il piano di progetto delinea le linee guida da seguire nell'organizzazione del team di sviluppo del progetto, nella descrizione di eventuali rischi, requisiti hardware e software, tempistica di rilascio dei vari step di sviluppo prevedendo anche le fasi di test. Il sistema dovrà aiutare l'utente nella consultazione di documenti quali referti online attraverso un'interfaccia web-based facilmente accessibile e utilizzabile.

2. organizzazione del progetto: descrive il modo in cui il team di sviluppo viene organizzato, le persone coinvolte e i loro ruoli;

Per lo sviluppo del sistema si è scelto di utilizzare uno sviluppo evolutivo (AGILE) in modo da garantire così un continuo aggiornamento tra l'azienda sanitaria e gli sviluppatori, verranno rilasciate delle bozze di interfacce e prototipi che serviranno a far capire al cliente in che direzione si sta andando.

- 1 Capo progetto: È il responsabile dell'intero team, si occupa di coordinare il lavoro, si occupa di dialogare con il committente, di gestire eventuali problemi architetturali e di sviluppo del progetto.

Nell'area di sviluppo ci saranno:

- 1 Analista software: il cui compito è analizzare ed interpretare le esigenze degli utenti, è incaricato della progettazione, codifica e documentazione, collaudo e manutenzione del sistema.

- 4 Programmatori web: hanno il compito di sviluppare il sito web con tutte le funzionalità.

- 1 Designer: ha il compito di rendere facilmente usabile e accessibile l'intero sistema.

Specifiche e obiettivi (quelli che ha inserito lui nel compito)

3. analisi del rischio: descrive i possibili rischi del progetto e la probabilità che si verifichino e propone le strategie di riduzione;

- È impossibile assumere uno staff con le abilità richieste per il progetto;
- Un membro chiave del gruppo si ammala in un momento critico per il progetto;
- Hardware essenziale per il progetto non viene consegnato in tempo;
- Il tempo per lo sviluppo del software è sottostimato;
- Cambiamento dei requisiti;
- Prolungamento dei tempi dovuto a nuovo ciclo di test;

4. requisiti di risorse hardware e software: specifica l'hardware e il software di supporto, richiesti per eseguire lo sviluppo.

Database;

Server web, con repository;

Protocolli di sicurezza;

Sito web;

Sviluppo usando applicativi IDE come Eclipse, SketchFlow per le bozze e i prototipi.

5. divisione del lavoro: delinea la divisione del progetto in attività e identifica le milestone e le consegne associate a ogni attività;

Task Work Package

T0 Analisi

T1 Piano di Progetto

T2 Analisi e Specifica dei Requisiti

M0 Colloquio con il cliente

T3 Organizzazione del lavoro

T4 Bozza sito web

M1 Convalida parziale con il cliente

T6 Implementazione del layout di base

T7 Implementazione delle funzionalità principali

M2 Convalida parziale con il cliente

T8 Testing funzionalità principali

T9 Miglioramento design grafico

T10 Implementazione delle funzionalità avanzate

M3 Convalida finale con il cliente

T11 Testing sito web

M4 Consegna versione finale del sito web

6. tempistica del progetto: mostra le dipendenze tra le attività, il tempo stimato richiesto per raggiungere ogni milestone e l'assegnazione delle persone alle attività;

Considerato un team composto come descritto nel paragrafo 2, e considerata una giornata lavorativa da 8 ore, la consegna definitiva è prevista in 200 giorni lavorativi.

2- Gestire le risorse umane, fare riferimento al People Capability Maturity Model P-CMM

P-CMM People Capability Maturity Model che può essere utilizzato come struttura per migliorare la gestione delle risorse umane. Fornisce uno schema basato su livelli di maturità crescenti dei processi per la gestione del personale.

Livello di maturità 1: Iniziale

Partendo dall'assunzione che una pratica non può essere migliorata se non è "ripetibile", il livello di maturità iniziale vede l'organizzazione effettuare la gestione delle persone tramite procedure ad hoc, spesso informali e non ripetibili se non sporadicamente. L'attività principale da compiere in questa fase è quella di aiutare l'organizzazione a rimuovere ogni impedimento alla "ripetibilità" delle pratiche.

Livello di maturità 2: Gestione

Al livello di maturità 2, l'organizzazione stabilisce una politica per divulgare presso tutti i gruppi di lavoro i processi stabiliti. Prima di pensare ad ogni miglioramento, l'organizzazione deve assicurare un ambiente di lavoro stabile in cui eseguire in maniera "ripetibile" i propri processi. A questo livello il management lascia ai responsabili dei singoli gruppi il compito di controllare il lavoro quotidiano, dedicandosi a sua volta al controllo dei risultati finali e della baseline (ed alle rispettive modifiche). Solo quando le pratiche stabilite saranno eseguite con naturalezza dall'intera organizzazione, questa potrà iniziare la fase successiva di utilizzo di processi comuni a tutta l'organizzazione.

Livello di maturità 3: Definizione

Al livello di maturità 3, l'organizzazione seleziona le migliori pratiche e le include in un processo comune. Operando tutti con le stesse pratiche definite, l'organizzazione sarà in grado di valutare le pratiche con migliori performance nell'ambiente comune. Documentate nell'ambito del processo comune le pratiche, queste diventano anche lo strumento di apprendimento per le nuove persone. Le misure effettuate sulle pratiche di maggiore criticità sono registrate in un archivio ed utilizzate per effettuarne l'analisi.

In tale modo si è creato il fondamento per una cultura di base comune all'organizzazione: un processo comune conosciuto ed applicato da tutti. E' il fondamento della cultura professionale di base dell'organizzazione.

Livello di maturità 4: Ripetibile

Al livello di maturità 4, l'organizzazione inizia a gestire i processi in base ai risultati utilizzando l'analisi delle misure effettuate. Le attività sono svolte secondo i processi comuni definiti ed i risultati sono quindi più "predicibili" in base all'esperienza storica. Le deviazioni dai risultati attesi sono analizzate, le cause delle deviazioni individuate e le azioni correttive prese di conseguenza.

I processi sono quindi gestiti quantitativamente ed i risultati sono "predetti" con maggiore cura. I risultati del business sono controllati dai "numeri" e non più dalle "milestone" come prima. Si crea quindi la cultura per un vero miglioramento dei processi e quindi delle performance reali.

Livello di maturità 5: Ottimizzazione

Al livello di maturità 5, l'organizzazione opera utilizzando in maniera ripetitiva i propri processi, ne valuta le performance quantitativamente ed opera per migliorarli di continuo. Gli eventuali difetti sono analizzati e le cause che li generano sono rimosse per evitare il loro ripetersi.

Le persone sono culturalmente abituate ad eseguire i processi conosciuti ed il management a gestirli quantitativamente ed a migliorarli. Si crea anche la cultura dell'accettazione del cambiamento. L'organizzazione entra in un circolo virtuoso di miglioramento continuo!

3- Differenze, vantaggi e svantaggi di Architetture a oggetti distribuiti e Architetture client-server

Architetture client-server:

L'applicazione viene modellata come un insieme di servizi forniti da server e un insieme di client che li utilizza.

I client devono conoscere i server disponibili, ma solitamente non sanno dell'esistenza di altri client.

Client e server sono processi separati. Diversi processi server possono essere eseguiti su un singolo processore server.

Architetture a oggetti distribuiti:

- limita la flessibilità: decidere dove bisogna fornire i servizi;
- i Client ricevono i servizi dai server (non da altri client);
- i server possono fungere da client ricevendo servizi da altri server, ma non possono richiedere servizi ai client;
- i client devono conoscere i servizi offerti da specifici server e devono sapere come contattarli;

- oggetti distribuiti: eliminare la distinzione tra client e server;

OGGETTI: i componenti fondamentali del sistema: Forniscono servizi & ricevono servizi da altri oggetti senza distinzione logica tra un client (recettore di un servizio) e un server (fornitore di servizio).

Gli oggetti possono essere distribuiti su diversi computer di una rete e comunicare attraverso un middleware chiamato ORB (object request broker).

Architetture a oggetti distribuiti: vantaggi

- permette ai progettisti del sistema di ritardare le decisioni su dove e come i servizi dovrebbero essere collocati, poiché gli oggetti che forniscono servizi possono essere posti su ogni nodo della rete;
- non c'è necessità di decidere in anticipo dove collocare gli oggetti della logica applicativa.
- è un'architettura di sistema molto aperta che permette di aggiungere nuove risorse quando vengono richieste.
- Il sistema è flessibile e scalabile.
- È possibile riconfigurare il sistema dinamicamente con oggetti che migrano attraverso la rete.
- Standard CORBA;

Architetture a oggetti distribuiti: svantaggi

- le prestazioni peggiorano se sono utilizzati molti oggetti piccoli o con servizi piccoli;
- maggior complessità rispetto ai sistemi client/server;

4- Sistemi di controllo (centralizzato, basato su eventi)

I modelli per strutturare un sistema si occupano della sua scomposizione in sottosistemi.

I modelli di controllo, a livello architetturale, si occupano del **flusso di controllo** tra i sottosistemi.

Due stili di controllo generici sono utilizzati nei sistemi software.

1. Controllo centralizzato: un sottosistema ha la responsabilità generale del controllo e avvia e ferma gli altri sottosistemi, può anche lasciare il controllo a un altro sottosistema, ma aspetterà di rientrarne in possesso.

2. Controllo basato su eventi: invece di integrare le informazioni di controllo in un unico sottosistema, ogni sottosistema può rispondere a eventi generati esternamente da altri sottosistemi o dall'ambiente del sistema. Ogni sottosistema è anche controllore del flusso.

Gli stili di controllo sono utilizzati in congiunzione con gli stili strutturali.

5- Milestone, tempistiche

Milestone:

- Report e documenti che descrivono lo stato del software che si sta sviluppando. dire come sta procedendo il lavoro, aggiornare le stime dei costi e la tempistica.
- Bisognerebbe stabilire una serie di **milestone** (letteralmente "pietre miliari"), ovvero punti "cardine" riconoscibili nell'attività del processo software.
- A ogni milestone dovrebbe esserci un output formale, come un report, che possa essere presentato al responsabile.
- I report delle milestone non devono essere dei documenti voluminosi ma un breve resoconto di cosa è stato completato.
- Le milestone dovrebbero rappresentare la fine di una fase separata e razionale del progetto.

Le consegne in genere sono delle milestone, ma le milestone non sono necessariamente delle consegne, possono essere anche risultati interni usati dal gestore del progetto per verificarne i progressi, ma non consegnati al cliente.

Tempistiche: il tempo e le risorse richiesti per completare le attività e organizzarle in una sequenza coerente. A meno che il progetto di cui si deve stimare la tempistica non sia simile a un progetto precedente, le valutazioni pregresse sono una base incerta. La stima della tempistica viene inoltre complicata dal fatto che progetti diversi possono usare metodi di progettazione e linguaggi di implementazione diversi. Quando un progetto è tecnologicamente avanzato, le stime iniziali saranno quasi certamente ottimistiche, anche quando si prova a considerare tutte le eventualità. Rispettando questo, la creazione di tempistiche del software non è molto diversa dalla creazione di tempistiche per altri tipi di progetti avanzati: nuovi velivoli, ponti e modelli di automobili sono spesso in ritardo a causa di problemi imprevisti.

Le tempistiche, quindi, devono essere aggiornate continuamente appena nuove informazioni sono disponibili.

La tempistica richiede la suddivisione dell'intero progetto in attività separate, e le stime di tempo richiesto le completano.

Grafo delle attività e diagramma di Gantt.