

PMP



Project Management Plan

Gruppo 7

Team Manager/Editore:

Roberto Di Russo

0510201086

robertodirusso@yahoo.it

Composizione del Team

| | | | |
|------------|-----------|------------|------------------------------|
| Maggiolini | Giovanni | 0510201080 | giannimag@hotmail.com |
| Annunziata | Umberto | 0510201230 | umberto287@libero.it |
| Nicolao | Francesco | 0510200141 | frc.nicolao@hotmail.it |
| Gargiulo | Claudio | 0510200990 | gargiulo.claudio@gmail.com |
| Iuliano | Vincenzo | 0510201428 | iuliano_vincenzo@hotmail.com |

Indice

1 Introduzione

| | | |
|-----|------------------------------|--------|
| 1.1 | Overview del Progetto | pag. 3 |
| 1.2 | Deliverables del Progetto | pag. 4 |
| 1.3 | Evoluzione del Progetto | pag. 5 |
| 1.4 | Materiale di Riferimento | pag. 5 |
| 1.5 | Definizioni ed Abbreviazioni | pag. 6 |

2 Organizzazione del Progetto

| | | |
|-----|----------------------------|---------|
| 2.1 | Modello del Processo | pag. 7 |
| 2.2 | Struttura Organizzativa | pag. 9 |
| 2.3 | Interfacce Organizzative | pag. 11 |
| 2.4 | Responsabilità di Progetto | pag. 12 |

3 Processi Gestionali

| | | |
|-----|--|---------|
| 3.1 | Obiettivi e Priorità | pag. 14 |
| 3.2 | Assunzioni, Dipendenze, Vincoli | pag. 14 |
| 3.3 | Gestione dei Rischi | pag. 15 |
| 3.4 | Meccanismi di Monitoraggio e Controllo | pag. 18 |
| 3.5 | Pianificazione dello Staff | pag. 19 |

4 Processi Tecnici

| | | |
|-----|--------------------------------------|---------|
| 4.1 | Metodi, Strumenti, Tecniche | pag. 20 |
| 4.2 | Documentazione del Software | pag. 21 |
| 4.3 | Funzionalità di Supporto al Progetto | pag. 22 |

5 Pianificazione del Lavoro, delle Risorse Umane e del Budget

| | | |
|-----|--|---------|
| 5.1 | Work Packages | pag. 23 |
| 5.2 | Dipendenze | pag. 32 |
| 5.3 | Risorse Necessarie | pag. 32 |
| 5.4 | Allocazione del Budget e delle Risorse | pag. 33 |
| 5.5 | Pianificazione | pag. 33 |
| 5.6 | Valutazione dello staff | pag. 34 |
| 5.7 | Meeting Agenda | pag. 38 |

1 Introduzione

1.1 Overview del Progetto

L'attività di gestione dei ricoveri rappresenta senza dubbio uno degli aspetti critici nell'organizzazione del lavoro di una struttura sanitaria che offre servizi di degenza ai suoi assistiti.

Le procedure ospedaliere orientate a tal proposito danno origine ad una mole di dati la cui gestione manuale risulta spesso onerosa in termini di tempo e che necessita di locali specifici per la conservazione dei documenti prodotti.

L'informatizzazione di tale attività permette un miglioramento sia in termini di tempo, potendo avere a disposizione copie multiple aggiornabili in pochi minuti, che in termini di spazio, potendo memorizzare il tutto in formato digitale.

Inoltre, la disponibilità di interfacce dal facile utilizzo, in concomitanza di adeguati livelli di sicurezza, permettono a chiunque fa parte dello staff ospedaliero di poter gestire una quota di dati, senza causare situazioni di inconsistenza e/o incoerenza.

Adeguati strumenti informatici dal facile utilizzo, quindi, possono sensibilmente contribuire ad una riduzione dei costi e dei tempi relativi alla gestione dei dati, che a sua volta comporta un miglioramento della qualità dell'assistenza offerta, obiettivo principale di qualsiasi struttura sanitaria.

Lo scopo di questo software è, dunque, quello di fornire un sistema semplice da utilizzare, veloce ed economico che garantisca efficienza nelle procedure di gestione dei ricoveri.

Il software prodotto sarà rivolto esclusivamente alle cliniche private, da qui il nome 'EasyClinique', e, di conseguenza, non si occuperà di casi di pronto soccorso.

1.2 Deliverables del Progetto

Ad inizio progettazione sono state fissate delle scadenze (deadlines) per la consegna dei documenti prodotti relativamente a ciascuna fase del ciclo di vita del software (deliverables).

Se ne riporta di seguito una panoramica.

- RAD (Documento di Raccolta e Analisi dei Requisiti)
 - versione 1.0 (fase di Raccolta) da rilasciare il 14/04/2008
 - versione 2.0 (fase di Analisi) da rilasciare il 23/04/2008
- SDD (Documento di System Design), la cui versione completa è da rilasciare il 05/05/2008
- ODD (Documento di Object Design), la cui versione completa è da rilasciare il 15/05/2008
- Implementazione del Sistema Software con Documento di Testing, le cui prime versioni sono da rilasciare il 25/05/2008
- Documentazione completa e finale e Piano di Progetto completo, da consegnare il 05/06/2008

Oltre alle versioni definitive appena elencate, sono previste anche delle versioni intermedie (documenti raw, grezzi), non consegnate direttamente al committente, ma gestite all'interno del team di sviluppo.

Ciascuno dei documenti di milestone viene rilasciato attraverso una sottomissione da parte del Team Manager nella sezione “Assignments” sul sito del corso di 'Ingegneria del Software – Cl. 1', aperto dal docente/committente sulla Running Platform della Facoltà di Scienze MM.FF.NN dell'Università di Salerno.

1.3 Evoluzione del Progetto

Il software EasyClinique, nella totalità dei documenti prodotti, è il risultato di un processo iterativo ed incrementale.

Si tratta, quindi, di un 'oggetto' in continua evoluzione.

Ogni documento, infatti, prima di essere consegnato al committente, viene più volte revisionato ed approvato in primo luogo dal Team Manager e, successivamente, da tutti i componenti del gruppo, in modo tale che ognuno degli sviluppatori possa dare il proprio contributo alla verifica del lavoro ed inoltre abbia una visione ben precisa di quella che è la configurazione attuale del progetto.

1.4 Materiale di Riferimento

Per la stesura dei documenti relativi a ciascuna fase del ciclo di vita di EasyClinique è stato consultato del materiale di riferimento, che viene di seguito elencato.

- Slides e dispense messe a disposizione dal docente, reperibili nella sezione “Risorse” del sito del corso di 'Ingegneria del Software – Cl. 1' della Runnig Platform
- Modelli di riferimento per i documenti da rilasciare, reperibili nella sezione “Risorse” del sito del corso di 'Ingegneria del Software – Cl. 1' della Runnig Platform
- “Object-Oriented Software Engineering – Using UML, Patterns and Java” (2nd edition) di Bernd Bruegge e Allen H. Dutoit, edito da Prentice Hall
- Materiale reperito in rete

1.5 Definizioni e abbreviazioni

Viene riportato in seguito un elenco degli acronimi e delle abbreviazioni utilizzate nel presente documento, seguiti dai rispettivi significati.

Si noti che il loro utilizzo viene fatto sia nella variante maiuscola, che in quella minuscola

- EasyClinique, nome scelto per il prodotto software
- RAD, Requirements Analysis Document, documento di raccolta ed analisi dei requisiti
- SDD, System Design Document, documento di design del sistema
- ODD, Object Design Document, documento di progettazione degli oggetti e dell'implementazione del sistema
- PMP, Project Management Planning, documento di pianificazione del progetto
- UML, Unified Modeling Language
- SQL, Structured Query Language
- JDBC, Java DataBase Connectivity
- DBMS, DataBase Management System
- Scienze MM.FF.NN., Scienze Matematiche Fisiche e Naturali
- RP, Running Platform
- DMI, Dipartimento di Matematica ed Informatica
- SW, software
- HW, hardware
- CVS, Ciclo di Vita del Software
- TM, Team Manager/Team Leader
- VTM, Vice-Team Manager

2 Organizzazione del Progetto

2.1 Modello del Processo

Per lo sviluppo del progetto si è scelto di gestire il CVS attraverso un Modello a Spirale, che prevede uno sviluppo iterativo ed incrementale.

Ciò comporta che l'intero processo di sviluppo è caratterizzato da una serie di versioni crescenti, cioè da elaborazioni sempre più dettagliate.

Come previsto dal suddetto modello, ad ogni iterazione vengono ripetute quattro fasi, dette anche task regions, che vengono di seguito riportate.

- **Determinazione di Obiettivi e Vincoli**

Gli obiettivi vengono definiti per acquisire più informazioni possibili relative alla particolare fase del progetto; i vincoli, invece, vengono stabiliti per capire le limitazioni legate alla fase stessa.

- **Valutazione dei Rischi e delle Alternative**

E' importante conoscere quali sono i rischi legati a ciascuna fase dello sviluppo, in quanto, spesso si pone la necessità di modificare le proprie azioni per ridurre la possibilità che si verifichino, riducendo, così, la probabilità che l'intero processo fallisca.

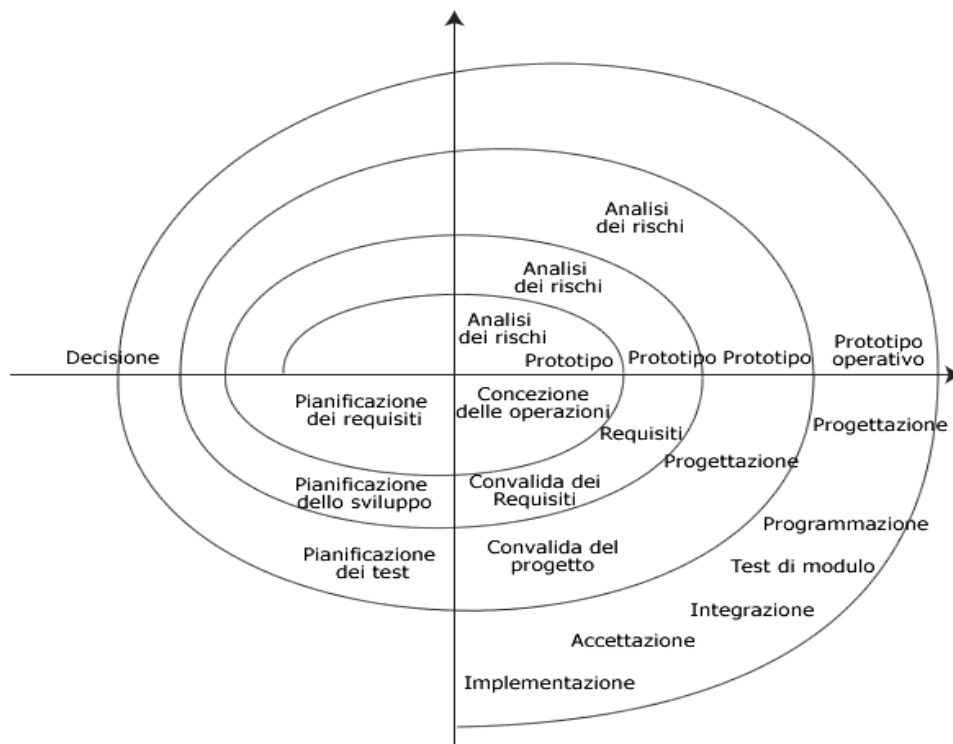
- **Sviluppo e Verifica:** viene scelto un modello per lo sviluppo del sistema nella fase in analisi e si procede allo sviluppo vero e proprio.

- **Pianificazione Fase Successiva:** viene revisionato il progetto e vengono prese delle decisioni sulle eventuali iterazioni successive.

La scelta di questo modello è stata effettuata in quanto è caratterizzato da una estrema flessibilità e per il fatto che esso racchiude le migliori caratteristiche degli altri modelli di CVS.

Tuttavia, la sua estrema flessibilità grava sui compiti del Team Manager, il quale è chiamato a mostrare una forte competenza nella gestione dei rischi per evitare problemi che potrebbero, addirittura, causare anche la riprogettazione e la reimplementazione del sistema, molto onerosa sia in termini di tempo che in termini di budget.

Si riporta, in seguito, una rappresentazione grafica del modello di CVS a Spirale



2.2 Struttura Organizzativa

La struttura organizzativa scelta per la gestione del team è il modello 'Controllato Decentralizzato', che prevede la presenza di un leader, appunto il Team Manager, che coordina il lavoro di subteams.

Tale modello prevede una comunicazione verticale dei subteams con il team manager, ed una comunicazione orizzontale tra i membri dei subteams.

Tuttavia, il Team Manager, per questo particolare progetto ha deciso applicare delle varianti al modello:

- non vengono nominati dei subleaders per i vari subteams,
- è possibile modificare la composizione dei sottoteams in ogni fase dello sviluppo

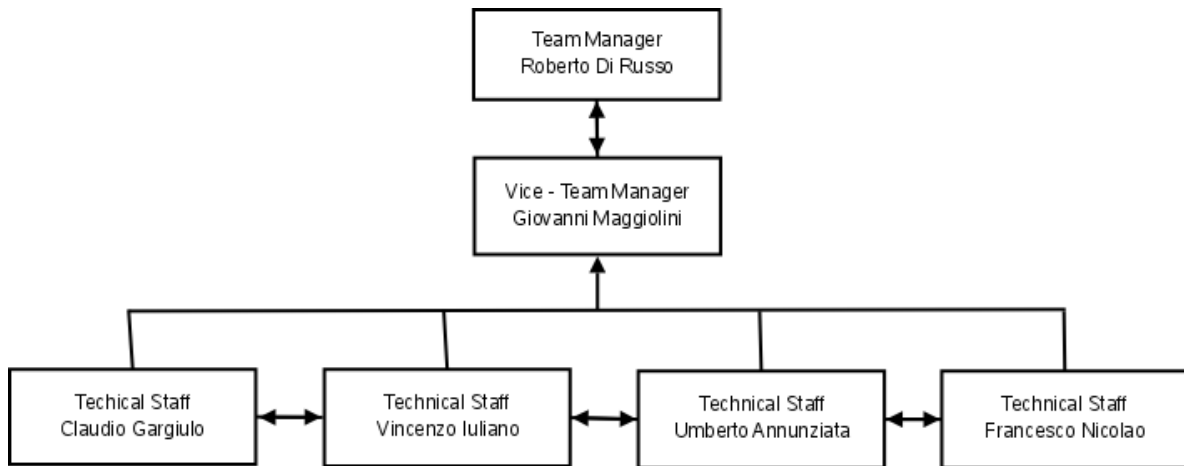
Tali scelte vengono effettuate dato l'esiguo numero di componenti del team (sei individui, compreso il leader) ed in modo da permettere ad ognuno di loro di lavorare a turno con un diverso 'collega'.

Si noti che spesso, in particolare durante la stesura del RAD e del SDD, il TM, oltre al suo, ricopre anche un ruolo da sviluppatore.

E' stato ritenuto opportuno da parte dello stesso TM di nominare un vice che possa essere di aiuto nella gestione del gruppo, ad esempio, nel periodo in cui lo stesso leader è impegnato nella stesura del presente documento o anche in fase di stesura dei documenti di Agenda.

Spesso, Manager e Vice-Manager compongono un subteams, lavorando insieme e calandosi anche nel ruolo di analisti/sviluppatori.

Viene riportata di seguito una rappresentazione grafica della struttura del team.



Come si evince dalla figura, non sono state definite in modo rigoroso le figure di Backup Engineer (BE), cioè colui che è incaricato di revisionare il progetto, e di Software Librarian (SL), ovvero colui che mantiene la documentazione storica del progetto.

Per quanto riguarda il BE, è stato concordato che, essendo un team alla prima esperienza, il ruolo verrà ricoperto a turno da ciascuno dei membri; mentre il ruolo di SL è ricoperto, sostanzialmente dal TM e dal VTM.

Si può notare, inoltre, che in figura non sono ben distinguibili i subteams: ciò avviene per i motivi precedentemente elencati.

2.3 Interfacce Organizzative

Le forme di relazione esterna che il team ha instaurato riguardano la comunicazione con il docente/committente (Professoressa Filomena Ferrucci), e con una persona impiegata presso una struttura ospedaliera privata ubicata nel comune di Formia (LT), che ricopre il ruolo di cliente (cioè fornisce, ad esempio, i desiderata che il prodotto SW deve soddisfare) e che per ovvie ragioni di privacy non viene nominata.

Per quanto riguarda la comunicazione con il cliente, l'unico membro ad avere rapporti diretti è il TM, essendo, tra l'altro l'unico a risiedere nel comune di ubicazione della struttura.

Essendo, comunque, un progetto a scopi didattici, il lavoro viene supervisionato periodicamente dal docente, mentre gli incontri con il personale della clinica sono stati necessari in fase di raccolta ed analisi dei requisiti alla comprensione del dominio del problema.

Si noti, quindi, che il cliente è uno pseudo-cliente: il suo ruolo è stato quello di esternare al team le richieste, ma in effetti, alla fine del progetto non riceve alcun documento e/o prodotto SW, anche perchè per il progetto attuale è prevista soltanto una implementazione parziale.

L'avvio del progetto è stata la fase più tortuosa.

In primo luogo il TM ha incontrato il contatto all'interno della struttura ospedaliera per definire una prima bozza delle specifiche del problema.

Già in questa fase, le difficoltà sono state non poche, in quanto, spesso, il committente ha mostrato insicurezza nelle richieste, e talvolta “pretendendo” una conoscenza del dominio tecnico, in questo caso medico-sanitario, da parte dell'interlocutore.

Si è svolto, poi, un incontro per l'approvazione da parte del docente della proposta di progetto (Project Proposal), elaborata in un primo momento, appunto, dal TM e raffinata con l'aiuto dell'intero team.

Successivamente, sono stati definiti il materiale da utilizzare per il supporto (di natura software in particolare) e le consegne iniziali.

La comunicazione con questa entità esterna è avvenuta e avviene per tutta la durata del CVS in diversi modi: dialogo diretto o indiretto e annunci sulla RP. Sulla stessa RP vengono consegnate le deliverables, attraverso la sottomissione da parte del TM (robertodirusso@yahoo.it) nella sezione “Assignments”.

Il ruolo del VTM è stato particolarmente utile nella comunicazione con il docente, secondo il principio “due menti sono meglio di una”, e nella chiarificazione di tutto ciò che il TM ha raccolto su carta nel dialogo con il cliente.

2.4 Responsabilità di Progetto

Vengono di seguito elencate le funzioni, cioè le attività che accompagnano il progetto durante l'intero arco di tempo del suo sviluppo.

- ***Analisi e gestione dei rischi***, l'identificazione dei problemi che possono avere effetti devastanti sulla costruzione del sistema e la ricerca di metodi per evitarli o nel caso si verifichino comunque, per risolverli.
Ne è responsabile il TM, supportato all'occorrenza dal VTM.
- ***Stesura della documentazione del progetto***, la raccolta degli elaborati prodotti dallo staff tecnico, la loro integrazione e presentazione.
La correzione, integrazione e revisione vengono fatte dal TM, mentre della formattazione (stili, font, allineamenti) ne è responsabile il VTM.
- ***Gestione del progetto***, di responsabilità del TM, prevede la coordinazione e la supervisione dell'intero processo di sviluppo.
- ***Controllo della disponibilità degli strumenti***, l'accertamento della disponibilità di materiale e strumenti nelle fasi di sviluppo del software.
Ne è responsabile il TM.
- ***Identificazione e perseguimento degli obiettivi***, prevede in ogni fase del CVS, l'identificazione degli obiettivi ed il relativo sviluppo.
Una prima proposta viene avanzata dal TM, supportato dal VTM, mentre poi è l'intero team a validare il target effettivo.
- ***Controllo della qualità***, una continua verifica del 'cosa' viene sviluppato e del 'come' lo si sviluppa. Ne è responsabile il TM.
- ***Configuration Management***, la conservazione dei documenti e degli altri prodotti sviluppati nella realizzazione del sistema. Ne sono responsabili il TM ed il VTM.

Volendo riassumere, i compiti principali del TM sono:

- dialogo con il cliente e con il docente;
- analisi e gestione dei rischi
- correzione, integrazione e revisione dei documenti prodotti dallo staff tecnico
- configuration management
- controllo qualità e disponibilità degli strumenti
- gestione progetto
- prima identificazione degli obiettivi di ciascuna fase del CVS

Il VTM ha, invece, il ruolo di supportare ed eventualmente supplire il TM e di mantenere una seconda copia delle configurazioni del sistema.

Gli altri componenti del team ricoprono principalmente i ruoli di sviluppatori, analisti e, a turno di Backup Engineers.

3 Processi Gestionali

3.1 Obiettivi e Priorità

Il target del sistema proposto e sviluppato è quello di facilitare e migliorare la gestione delle degenze presso una struttura ospedaliera privata, nonché di fornire, in primo luogo al dirigente sanitario della stessa, una serie di utilità mirate alla gestione del personale e delle statistiche relative al lavoro svolto.

Fornendo adeguati strumenti informatici dal facile utilizzo, quindi, si può sensibilmente contribuire ad una riduzione dei costi e dei tempi relativi alla gestione dei dati, che a sua volta comporta un miglioramento della qualità dell'assistenza offerta da parte della struttura sanitaria ai suoi assistiti.

Lo scopo di questo software è, dunque, quello di fornire un sistema semplice da utilizzare, veloce ed economico che garantisca efficienza nelle procedure di gestione dei ricoveri.

Ovviamente, l'intero team si impegna nel rispettare rigorosamente i tempi di consegna stabiliti, dettati dal docente, più che dal cliente, senza che ciò vada oltremodo ad influire sulla qualità dei documenti prodotti.

3.2 Assunzioni, Dipendenze, Vincoli

I vincoli imposti per lo sviluppo del sistema SW sono l'uso di UML (Unified Modeling Language), standard internazionale previsto nella rappresentazione diagrammatica come linguaggio di modellazione.

Inoltre, è stato richiesto l'utilizzo di un linguaggio di programmazione ad oggetti nella realizzazione del sistema SW. In particolare, è stato scelto il linguaggio Java 5.0 della Sun Microsystem, essendo l'unico linguaggio object-oriented con cui l'intero team ha dimestichezza.

Il docente pretende un impegno full-time dei partecipanti alla realizzazione del progetto, anche se questo vincolo è in parte ostacolato dagli impegni accademici (corsi, prove intercorso ed altro) in cui sono impegnati i componenti del team.

In data 05/03/2008, con ben dieci giorni di anticipo rispetto alla data di inizio del progetto, viene delineata la composizione del team:

- i. Roberto Di Russo (RDR) – Team Manager
- ii. Giovanni Maggiolini (GM)
- iii. Umberto Annunziata (UA)
- iv. Francesco Nicolao (FN)
- v. Claudio Gargiulo (CG)

In data 10/03/2008, quindi con cinque giorni di anticipo rispetto alla data di inizio del progetto, viene aggiunta un'altra risorsa umana al team:

- vi. Vincenzo Iuliano (VI)

Parallelamente, il TM nomina come suo Vice Giovanni Maggiolini

3.3 Gestione dei Rischi

Qualsiasi progetto SW può essere soggetto a problemi che possono venire fuori nelle fasi iniziali oppure, con maggiore probabilità, nel corso dello sviluppo del sistema.

Tali problemi possono essere tali da portare ad una reimplementazione del sistema o addirittura ad un abbandono di esso, nel caso in cui richieda uno sforzo troppo oneroso in termini di tempo e risorse.

Una corretta individuazione e gestione dei rischi, tuttavia, può scongiurare situazioni di questo genere.

Viene di seguito riportata una panoramica dei rischi individuati, ognuno corredato da un livello di gravità e dalla probabilità che si verifichi.

Notazione: G = Gravità (da 0 a 5)

P = Probabilità (da 0 a 3)

| Rischio | G | P | Descrizione | Gestione |
|---|----------|----------|--|--|
| Difficoltà di comunicazione all'interno del gruppo | 5 | 1 | I membri del gruppo non riescono a comprendersi I metodi adottati sono poco adatti alla comunicazione | Il TM organizza un training informale prima del progetto e stabilisce i metodi di comunicazione impiegati, cercando l'accordo da parte di tutto il team. |
| Membro assenteista e spesso irreperibile | 4 | 2 | Uno membro del gruppo è spesso assente ai meeting del progetto e non è reperibile dagli altri membri. | Dopo una prima ammonizione, il TM attua misure drastiche nei confronti del membro assenteista, procedendone, se ritenuto necessario all'esclusione dal progetto. |
| Abbandono del progetto da parte di un membro | 5 | 1 | Un membro del gruppo abbandona il progetto | Il TM provvede ad distribuire il lavoro del "dimissionario" tra i restanti membri del gruppo. |
| Membro non adeguatamente preparato | 3 | 3 | Un membro del gruppo non ha le competenze necessarie per partecipare in modo produttivo alle attività | Il TM cerca di affiancare il membro impreparato ad uno dei membri tecnicamente più competenti, facendo attenzione che ciò non porti rallentamenti eccessivi |
| Membro non disponibile per un periodo di tempo | 2 | 4 | Un membro del gruppo non può partecipare al progetto per un periodo di breve durata. | Il TM cerca di assegnare comunque un lavoro da svolgere in modo decentrato al membro interessato ed eventualmente provvede ad assegnarne temporaneamente i compiti in modo distribuito ai restanti membri del team |
| Scarsa attitudine al lavoro di gruppo da parte di un membro | 5 | 2 | Un membro del gruppo non riesce a lavorare in un contesto collettivo | Il TM, supportato dal VTM, cerca di fare in modo che il membro interessato si senta a suo agio all'interno del team, affiancandolo al membro più carismatico |
| Tempi di consegna non rispettati | 5 | 1 | Un deliverable non viene consegnato entro la data stabilita dal docente | Il gruppo cerca di evitare ritardi nella realizzazione del SW, ricorrendo anche a duplicare le ore di lavoro. |
| Scarsa comprensione dei requisiti dettati dal cliente | 5 | 2 | Il cliente non esprime le proprie richieste comprensibile | Il TM propone al cliente un numero maggiore di incontri ed elabora una serie di checklists per aiutarlo a definire in modo chiaro le specifiche |

| | | | | |
|---|----------|----------|---|---|
| Cambio dei requisiti da parte del cliente | 5 | 3 | Il cliente definisce nuovi requisiti per il sistema o richiede la modifica di alcuni requisiti | Il TM traghetta il gruppo nell'adattare il lavoro svolto, senza che ciò comporti cambiamenti drastici. |
| Scarsa sincronizzazione del progetto con altri corsi di studio | 5 | 3 | I componenti del gruppo non riescono a partecipare in modo proficuo al progetto e ad altri corsi di studio | Nel peggiore dei casi, i componenti sono costretti ad abbandonare almeno uno dei suddetti corsi e, di conseguenza, a rimandarne l'esame |
| Inesperienza del TM | 5 | 3 | Il TM non risulta abbastanza maturo da poter dirigere il team di sviluppo | Il TM viene supportato dal VTM, in modo da gestire il gruppo in modo ragionevole e nel frattempo maturare un minimo di esperienza per garantire il raggiungimento dei targets |
| Difficoltà nel dividere il sistema in sottosistemi | 3 | 2 | Il gruppo ha difficoltà nello scomporre il sistema in sottosistemi, basandosi sulle informazioni del RAD e le decisioni prese nel SDD | Nel caso non ci sia un rimedio per superare le difficoltà, si procede alla ridefinizione degli obiettivi di design e alla revisione dell'intero lavoro svolto |
| Ambiente di lavoro poco confortevole | 2 | 3 | L'ambiente di lavoro (aule universitarie) risulta dispersivo poco adatto alla concentrazione | Il TM manda in avanscoperta i membri del TM, in modo da trovare luoghi più tranquilli e silenziosi, all'interno della struttura universitaria |

3.4 Meccanismi di Monitoraggio e Controllo

Oltre ai meeting tenuti prima di ogni fase del progetto, il TM ha previsto all'inizio dello stesso che il gruppo aggiunga alle ore di lavoro individuali di ciascun membro una serie di incontri, da svolgersi in sede universitaria, in cui poter lavorare insieme, confrontarsi e favorire l'interazione tra le risorse umane. Questi incontri, seppur spesso di solo una o due ore hanno praticamente una cadenza giornaliera, seppur spesso sono tenuti in orari scomodi, a causa del poco tempo a disposizione dovuto al parallelismo con altri corsi del Piano di Studi.

Ogni giorno dedicato al progetto, il TM controlla ciò che è stato svolto da ciascun partecipante.

Quando richiesto, ogni membro del gruppo è tenuto a mostrare il lavoro svolto. Solitamente ciò viene fatto direttamente sul PC portatile del membro stesso, ma in casi di mancanze, viene utilizzata una penna USB di proprietà del TM.

A causa dell'impossibilità da parte del TM di collegarsi ad Internet, l'invio del lavoro tramite posta elettronica è previsto solo nel fine settimana.

Ogni membro deve, comunque, rispettare le scadenze interne previste dal TM. Per quanto riguarda i verbali dei meeting, dato l'elevato numero di incontri previsti, il TM ha deciso di stendere i documenti di Agenda solo nel primo giorno dedicato ad ogni fase od eventualmente in seguito a modifiche drastiche. Sarebbe di fatto impossibile, causa le imminenti scadenze e l'impegno richiesto da altri corsi stendere un verbale ogni volta che il gruppo si riunisce.

3.5 Pianificazione dello Staff

Il TM richiede ai componenti del suo staff una buona dimestichezza con UML, requisito fondamentale alla stesura dei documenti di milestone.

In particolare, i partecipanti al progetto devono possedere una buona conoscenza di Use Case Diagram, Sequence Diagram, Class Diagram, Object Model.

Ovviamente, oltre a queste conoscenze di base, è richiesta a ciascun membro la partecipazione al corso di Ingegneria del Software, nonché una discreta padronanza degli argomenti trattati nello stesso.

Per quanto riguarda i tools di modellazione, il TM non pretende dal suo staff una conoscenza di particolari strumenti, ma ha previsto nei primi incontri la definizione dei programmi da utilizzare ed un piccolo training su di essi.

Diverso discorso viene fatto per la parte implementativa del codice.

All'intero staff viene richiesta una più che buona padronanza dello stand-alone Eclipse per la programmazione Java, di cui si dà per assodata la conoscenza, essendo il superamento dell'esame di “Linguaggi di Programmazione II” una conoscenza pregressa indispensabile ai fini del progetto.

Sul piano tecnico, i requisiti pregressi ritenuti indispensabili per chiunque partecipi allo sviluppo sono:

- a) come detto, una buona conoscenza del paradigma object-oriented ed in particolare del linguaggio Java;
- b) buona conoscenza del linguaggio di interrogazione SQL e del DBMS MySQL;
- c) conoscenza del paradigma procedurale, senza vincoli di linguaggio;
- d) conoscenza di JDBC per l'interfacciamento Java – DBMS.

Su quest'ultimo punto, tuttavia, l'intero team compreso il TM, ha inizialmente presentato delle notevoli lacune. L'interesse del TM è stato quello di documentarsi e di diffondere il materiale reperito tra i partecipanti al progetto. Seppur con ritardo rispetto all'inizio della fase implementativa, il team è stato supportato sull'argomento anche da una lezione specifica tenuta durante l'orario del corso di “Ingegneria del Software” dal Dott. Carmine Gravino del DMI dell'Università di Salerno.

4. Processi Tecnici

4.1 Metodi, Strumenti e Tecniche

Le linee guida per la gestione e lo sviluppo del progetto sono quelle apprese al corso di Ingegneria del Software e dettate dal libro “Object-Oriented Software Engineering” di Brugge e Dutoit.

Per lo sviluppo del prodotto software in questione, ciascun membro del team utilizza il proprio PC portatile e/o i computer fissi messi a disposizione dall'Università di Salerno per l'implementazione del sistema, nonché per la stesura dei documenti.

Per quanto riguarda i tools di supporto, viene di seguito fornito un elenco dei prodotti utilizzati, ognuno corredato di una breve descrizione.

- Open Office.org Writer oppure Microsoft Word 2007: word processors utilizzati per stendere la documentazione di supporto al sistema sviluppato
- ArgoUml : in un primo momento utilizzato per disegnare i diagrammi UML, ma subito accantonato per la sua scarsa usabilità e per le poche funzioni messe a disposizione.
- Dia: utilizzato in sostituzione di ArgoUML per disegnare i diagrammi UML.
- Microsoft Project 2007: utilizzato per formalizzare la gestione delle risorse umane e le attività del progetto (utilizzato solo dal TM).
- Open Office.org Impress oppure Microsoft PowerPoint 2007: utilizzati per preparare le presentazioni
- Eclipse: utilizzato per l'implementazione (codice) e la stesura della javadoc
- modulo JDBC: utilizzato per realizzare il database
- MySql 5.0: utilizzato come DBMS

Si noti come si abbia cercato di utilizzare il più possibile strumenti freeware per garantire la portabilità degli elaborati, requisito necessario, dato che all'interno del team vengono utilizzati diversi sistemi operativi (Windows XP, Windows Vista, GNU/Linux nelle distribuzioni Ubuntu 7.10 e openSUSE 10.3, Mac OS X).

In sostituzione, sono stati utilizzati prodotti Microsoft.

4.2 Documentazione del Software

Ogni fase di sviluppo del software ha come output un documento, il cui scopo è quello di consentire il monitoraggio del lavoro svolto dal team, per valutarne completezza, consistenza, chiarezza e concretezza.

Tecnicamente si parla di milestones (fasi con documenti in output) e deliverables (documenti stessi).

Viene di seguito riportato un elenco descrittivo dei documenti prodotti durante le varie fasi dello sviluppo.

| Nome | Significato | Descrizione | Durata |
|-------------|------------------------------------|---|--|
| PMP | Project Management Planning | Documento di pianificazione del progetto, cioè il documento che si sta leggendo in questo momento. | Tutto il processo di sviluppo del software |
| RAD | Requirements and Analysis Document | Documenti di raccolta e analisi dei requisiti, di cui sono previste due consegne: una al termine della fase di raccolta, l'altro al termine della fase di analisi | Fasi di raccolta e analisi dei requisiti, con possibilità di revisione |
| SDD | System Design Document | Documento di System design | Fase di progettazione del sistema, con possibilità di revisione |
| ODD | Object Design Document | Documento di object design, preliminare dell' implementazione | Fase di object design, con possibilità di revisione |
| TP | Test Planning | Documento di pianificazione della fase di testing | Fase pre-Testing |
| TCS | TestCase Specification | Documento di specifica dei casi di test | Fase pre-Testing |
| TL | Test Log | Documento usato per tenere traccia dei risultati del testing | Fase di Testing |
| TSM | Test Summary Report | Documento usato per dare una valutazione finale al testing | Fase post-Testing |

Per il dettaglio del Revision History di ciascun prodotto , si rimanda alla consultazione di documenti sopra elencati.

4.3 Funzionalità di Sviluppo al Progetto

La Gestione delle Configurazioni, funzionalità che permette di gestire le evoluzioni dello sviluppo del progetto, è di competenza del TM.

Questa “attività” accompagna il progetto dal suo inizio al suo completamento e viene documentata inserendo all’inizio di ogni documento la sua Revision History.

Il suo compito è sostanzialmente quello di tenere aggiornato l'intero team sulla versione corrente di ciascun documento, alla quale si dovrà fare riferimento nelle fasi successive.

Nella cura di questa funzione il TM viene supportato dal VTM, il cui compito è quello di mantenere una copia aggiornata di ciascuna configurazione.

Si noti che per comodità e per non creare confusione tra i membri del team di sviluppo, è stato deciso di sovrascrivere ogni volta la versione obsoleta di un documento con quella aggiornata.

La fine del progetto vedrà, quindi, una sola versione per ogni documento, mentre per tenere traccia delle evoluzioni si farà riferimento, come detto, al Revision History.

5. Pianificazione del Lavoro, delle Risorse Umane e del Budget

5.1 Work Packages

Il processo di sviluppo di EasyClinique viene accompagnato da una assidua attività di Project Management, svolta, appunto, dal TM.

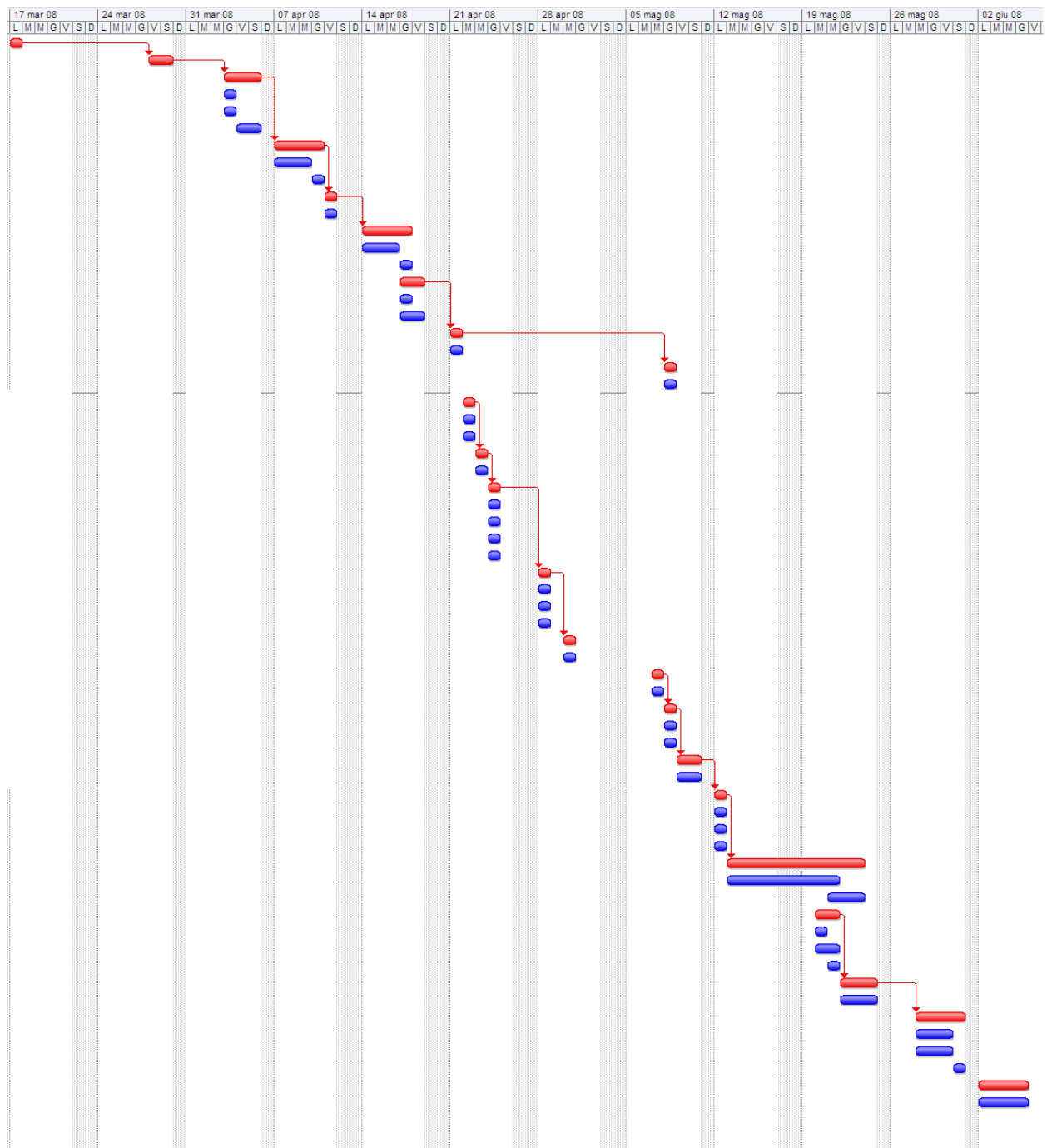
In seguito viene riportato lo scheduling delle attività, rappresentato in modo tabellare ed elaborato con Microsoft Project 2007.

| ID | Task Name | Durata | Inizio | Fine | Predecessori |
|----|---|-----------|--------------|--------------|--------------|
| 1 | Project Proposal 1.0 | 1 giorno | lun 17/03/08 | lun 17/03/08 | |
| 2 | Project Proposal 2.0 | 2 giorni | ven 28/03/08 | sab 29/03/08 | 1 |
| 3 | RAD 0.1 | 3 giorni | gio 03/04/08 | sab 05/04/08 | 2 |
| 4 | Stesura dell'introduzione | 1 giorno? | gio 03/04/08 | gio 03/04/08 | |
| 5 | Definizione dei requisiti funzionali e non | 1 giorno? | gio 03/04/08 | gio 03/04/08 | |
| 6 | Definizione degli scenari | 2 giorni? | ven 04/04/08 | sab 05/04/08 | |
| 7 | RAD 0.2 | 4 giorni | lun 07/04/08 | gio 10/04/08 | 3 |
| 8 | Definizione degli use cases | 3 giorni | lun 07/04/08 | mer 09/04/08 | |
| 9 | Definizione use case diagrams | 1 giorno? | gio 10/04/08 | gio 10/04/08 | |
| 10 | RAD 1.0 | 1 giorno | ven 11/04/08 | ven 11/04/08 | 7 |
| 11 | Integrazione e correzione lavoro svolto | 1 giorno? | ven 11/04/08 | ven 11/04/08 | |
| 12 | RAD 1.1 | 4 giorni? | lun 14/04/08 | gio 17/04/08 | 10 |
| 13 | Sviluppo diagrammi di sequenza | 3 giorni? | lun 14/04/08 | mer 16/04/08 | |
| 14 | Raffinazione use cases | 1 giorno? | gio 17/04/08 | gio 17/04/08 | |
| 15 | RAD 1.2 | 2 giorni? | ven 18/04/08 | lun 21/04/08 | 12 |
| 16 | Sviluppo Statecharts | 1 giorno? | gio 17/04/08 | gio 17/04/08 | |
| 17 | Sviluppo Diagramma Classi | 2 giorni? | gio 17/04/08 | ven 18/04/08 | |
| 18 | RAD 2.0 | 1 giorno? | mar 22/04/08 | mar 22/04/08 | 15 |
| 19 | Correzione ed integrazione del lavoro svolto | 1 giorno? | lun 21/04/08 | lun 21/04/08 | |
| 20 | RAD 3.0 | 1 giorno? | gio 08/05/08 | gio 08/05/08 | 18 |
| 21 | Aggiunta use case diagrams per packages | 1 giorno? | gio 08/05/08 | gio 08/05/08 | |
| 22 | SDD 0.1 | 1 giorno | mar 22/04/08 | mar 22/04/08 | |
| 23 | Introduzione | 1 giorno? | mar 22/04/08 | mar 22/04/08 | |
| 24 | Definizione System goals | 1 giorno? | mar 22/04/08 | mar 22/04/08 | |
| 25 | SDD 0.2 | 1 giorno | mer 23/04/08 | mer 23/04/08 | 22 |
| 26 | Decomposizione del sistema in sottosistemi | 1 giorno? | mer 23/04/08 | mer 23/04/08 | |
| 27 | SDD 0.3 | 1 giorno | gio 24/04/08 | gio 24/04/08 | 25 |
| 28 | Gestione Dati Persistenti | 1 giorno? | gio 24/04/08 | gio 24/04/08 | |
| 29 | Controllo accessi e sicurezza | 1 giorno? | gio 24/04/08 | gio 24/04/08 | |
| 30 | Gestione controllo globale | 1 giorno? | gio 24/04/08 | gio 24/04/08 | |
| 31 | Boundary Conditions | 1 giorno? | gio 24/04/08 | gio 24/04/08 | |
| 32 | SDD 1.0 | 1 giorno | lun 28/04/08 | lun 28/04/08 | 27 |
| 33 | Servizi dei sottosistemi | 1 giorno? | lun 28/04/08 | lun 28/04/08 | |
| 34 | Glossario | 1 giorno? | lun 28/04/08 | lun 28/04/08 | |
| 35 | Revisione ed integrazione del lavoro svolto | 1 giorno? | lun 28/04/08 | lun 28/04/08 | |
| 36 | SDD 1.1 | 1 giorno? | mer 30/04/08 | mer 30/04/08 | 32 |
| 37 | Inserimento dello schema ER nella Gestione Dati Persi | 1 giorno? | mer 30/04/08 | mer 30/04/08 | |
| 38 | ODD 0.1 | 1 giorno? | mer 07/05/08 | mer 07/05/08 | |
| 39 | Definizione della parte di sistema da implementare | 1 giorno? | mer 07/05/08 | mer 07/05/08 | |
| 40 | ODD 0.2 | 1 giorno? | gio 08/05/08 | gio 08/05/08 | 38 |
| 41 | Definizione dei packages | 1 giorno? | gio 08/05/08 | gio 08/05/08 | |
| 42 | Compromessi, Standard adottati, Acronimi ed abbrevia | 1 giorno? | gio 08/05/08 | gio 08/05/08 | |

PMP EasyClinique Gruppo 7

| ID |  | Task Name | Durata | Inizio | Fine | Predecessori |
|----|---|---|-----------|--------------|--------------|--------------|
| 43 |  | ODD 0.3 | 2 giorni? | ven 08/05/08 | sab 10/05/08 | 40 |
| 44 |  | Definizione delle interfacce delle classi | 2 giorni? | ven 08/05/08 | sab 10/05/08 | |
| 45 |  | ODD 1.0 | 1 giorno? | lun 12/05/08 | lun 12/05/08 | 43 |
| 46 |  | Precondizioni, postcondizioni, invarianti | 1 giorno? | lun 12/05/08 | lun 12/05/08 | |
| 47 |  | Glossario | 1 giorno? | lun 12/05/08 | lun 12/05/08 | |
| 48 |  | Revisione ed integrazione del lavoro svolto | 1 giorno? | lun 12/05/08 | lun 12/05/08 | |
| 49 |  | Implementazione | 9 giorni? | mar 13/05/08 | ven 23/05/08 | 45 |
| 50 |  | Effettiva realizzazione delle classi Java | 7 giorni? | mar 13/05/08 | mer 21/05/08 | |
| 51 |  | Verifica del software realizzato | 3 giorni? | mer 21/05/08 | ven 23/05/08 | |
| 52 |  | Test Planning | 2 giorni? | mar 20/05/08 | mer 21/05/08 | |
| 53 |  | Stesura della parte introduttiva | 1 giorno? | mar 20/05/08 | mar 20/05/08 | |
| 54 |  | Definizione dei test cases | 2 giorni? | mar 20/05/08 | mer 21/05/08 | |
| 55 |  | Pianificazione del testing | 1 giorno? | mer 21/05/08 | mer 21/05/08 | |
| 56 |  | TestCaseSpecification | 3 giorni? | gio 22/05/08 | sab 24/05/08 | 52 |
| 57 |  | Specifica dei test cases | 3 giorni? | gio 22/05/08 | sab 24/05/08 | |
| 58 |  | Testing | 4 giorni? | mer 28/05/08 | sab 31/05/08 | 56 |
| 59 |  | Attività di Testing e correzione dell'implementazione | 3 giorni? | mer 28/05/08 | ven 30/05/08 | |
| 60 |  | Stesura del TestLog | 3 giorni? | mer 28/05/08 | ven 30/05/08 | |
| 61 |  | Stesura del Test Summary Report | 1 giorno? | sab 31/05/08 | sab 31/05/08 | |
| 62 |  | Revisione Finale | 4 giorni? | lun 02/06/08 | gio 05/06/08 | |
| 63 |  | Revisione finale di tutti i documenti | 4 giorni? | lun 02/06/08 | gio 05/06/08 | |

Segue il diagramma di Gantt, in modo da offrire una rappresentazione grafica dello scheduling appena proposto.



NOTA: Ciascun rettangolo rosso rappresenta la stesura di una versione di documento (un'attività)
 Ogni rettangolo blu, invece, rappresenta un task.
 Si deduce facilmente che un rettangolo rosso è formato da un insieme di rettangoli blu.

Di seguito viene riportata la suddivisione dei task tra i membri del team

Generale

| Attività: | Realizzato da: |
|------------------------|----------------|
| Revisione | RDR, GM |
| Impaginazione | GM |
| Logo | CG |
| Project Proposal | RDR |
| Giustificazione scelte | RDR |

RAD

| #scenario. Usecase di riferimento | realizzato da: |
|-----------------------------------|----------------|
| 00. Login | RDR |
| 01. Logout | RDR |
| 02. ConsultazioneManuale | RDR |
| 03. RicercaPaziente | GM |
| 04. VisualizzazioneGrafico | CG |
| 05. InserimentoPersonale | FN |
| 06. ModificaPersonale | FN |
| 07. CancellazionePersonale | FN |
| 08. RicercaPersonale | FN |
| 09. StampaPersonale | VI |
| 10. InserimentoReparto | GM |
| 11. RicercaReparto | GM |
| 12. CancellazioneReparto | GM |
| 13. InserimentoStanza | CG |
| 14. RicercaStanza | CG |
| 15. ModificaStanza | CG |
| 16. InserimentoAccount | UA |
| 17. ModificaAccount | UA |
| 18. CancellazioneAccount | UA |
| 19. AssegnazionePostoLetto | RDR |
| 20. CambioReparto | RDR |
| 21. DimissionePaziente | VI |
| 22. InserimentoPaziente | GM |
| 23. ModificaPaziente | GM |
| 24. InserimentoCartellaClinica | UA |
| 25. ModificaCartellaClinica | UA |
| 26. StampaCartellaClinica | VI |
| 27. NotificaDimissione | VI |

| use case | realizzato da: |
|----------------------------|----------------|
| InserimentoStanza | CG |
| ModificaStanza | CG |
| RicercaStanza | CG |
| VisualizzazioneGrafico | CG |
| CancellazionePersonale | FN |
| InserimentoPersonale | FN |
| ModificaPersonale | FN |
| RicercaPersonale | FN |
| CancellazioneReparto | GM |
| InserimentoPaziente | GM |
| InserimentoReparto | GM |
| ModificaPaziente | GM |
| RicercaPaziente | GM |
| RicercaReparto | GM |
| ConsultazioneManuale | RDR |
| Login | RDR |
| Logout | RDR |
| AssegnazionePostoLetto | RDR |
| CambioReparto | RDR |
| CancellazioneAccount | UA |
| InserimentoAccount | UA |
| ModificaAccount | UA |
| InserimentoCartellaClinica | UA |
| ModificaCartellaClinica | UA |
| StampaCartellaClinica | VI |
| StampaPersonale | VI |
| DimissionePaziente | VI |
| NotificaDimissione | VI |

| |
|--------------------------|
| Diagrammi realizzati da: |
| Semplici: RDR e GM |
| Complessi: FN, UA, CG |
| Per Packages: VI |

| Sequence Diargams | realizzato da: |
|----------------------------|----------------|
| InserimentoStanza | CG |
| ModificaStanza | CG |
| RicercaStanza | CG |
| VisualizzazioneGrafico | CG |
| CancellazionePersonale | FN |
| InserimentoPersonale | FN |
| ModificaPersonale | FN |
| RicercaPersonale | FN |
| CancellazioneReparto | GM |
| InserimentoPaziente | GM |
| InserimentoReparto | GM |
| ModificaPaziente | GM |
| RicercaPaziente | GM |
| RicercaReparto | GM |
| ConsultazioneManuale | RDR |
| Login | RDR |
| Logout | RDR |
| AssegnazionePostoLetto | RDR |
| CambioReparto | RDR |
| CancellazioneAccount | UA |
| InserimentoAccount | UA |
| ModificaAccount | UA |
| InserimentoCartellaClinica | UA |
| ModificaCartellaClinica | UA |
| StampaCartellaClinica | VI |
| StampaPersonale | VI |
| DimissionePaziente | VI |
| NotificaDimissione | VI |

| |
|---|
| Diagrammi realizzati da: |
| Il realizzatore del diagramma è lo stesso della descrizione |

| cosa: | realizzato da: |
|----------------------|----------------|
| Introduzione | RDR |
| Sistema Proposto | RDR, GM |
| StateCharts Diagrams | UA |
| Modello delle Classi | RDR, GM, FN |
| Mock-up | CG |
| Glossario | VI |

SDD

| Attività | Svolta da: |
|-------------------------------------|-----------------|
| Introduzione e Overview del sistema | RDR |
| Definizione dei Design Goals | RDR, GM, VI, FN |
| Abbreviazioni e Riferimenti | RDR |
| Architettura del sistema attuale | RDR |
| Decomposizione in sottosistemi | GM |
| | CG |
| | UA |
| | GM |
| Hardware/Software Mapping | VI |
| Gestione Dati Persistenti | VI |
| Controllo degli Accessi e Sicurezza | GM |
| Gestione Controllo Globale | GM |
| Boundary Conditions | FN, CG |
| Servizi dei sottosistemi | UA, CG |
| Glossario | VI |

SottosistemaAmministratore
 SottosistemaMedicoParamedico
 SottosistemaResponsabile
 SottosistemaDirigente

ODD

| Attività | Realizzato da: |
|--|-------------------------|
| Definizione parte del sistema da implementare | RDR, GM, VI, UA, CG, FN |
| Compromessi, standard adottati, acronimi e definizioni | VI, RDR |
| Definizione dei packages | RDR, GM, VI, UA, CG, FN |
| Definizione delle interfacce delle classi | |
| Paziente, DBPaziente | GM |
| CartellaClinica, DBCartellaClinica | VI |
| Account, DBAccount | FN |
| Degenza, DBDegenza | UA |
| Glossario | VI |
| Precondizioni, postcondizioni, invarianti | CG |

Implementazione

| Classe | Realizzata da: |
|---|----------------|
| Paziente | GM |
| DBPaziente | GM |
| CartellaClinica | VI |
| DBCartellaClinica | VI |
| Degenza | UA |
| DBDegenza | UA |
| Account | FN |
| DBAccount | FN |
| Classi rappresentanti l'interfaccia grafica | CG, GM |

NOTA: Durante l'implementazione, RDR stende il PMP

| Attività | Realizzato da: |
|----------------|----------------|
| Manuale Utente | VI |

TestingPlanning

| Cosa: | Realizzato da: | |
|--|----------------|--------------------------------|
| Introduzione | VI, RDR | |
| TestCases | VI | VisualizzazioneCartellaClinica |
| | VI | ModificaCartellaClinica |
| | UA | InserimentoCartellaClinica |
| | UA | DimissionePaziente |
| | UA | AssegnazionePostoLetto |
| | UA | CambioReparto |
| | FN | Login |
| | FN | VisualizzazionePaziente |
| | FN | InserimentoPaziente |
| | FN | ModificaPaziente |
| Pianificazione | RDR, VI | |
| Determinazione Ruoli/Rischi | RDR, VI | |
| Decomposizione gerarchica del sistema | VI | |
| Schedulazione delle attività di testing | RDR | |
| Organizzazione delle attività di testing | RDR | schemi realizzati da VI |
| Glossario | VI | |

NOTA1: gli autori dei TestCases nel TP, quelli dei TestCases nel TestCaseSpecification e quelli dei TestLog n
TestLog coincidono

NOTA2: durante questa fase CG e GM sono impegnati nell'implementazione delle interfacce

TestingLog

| | |
|--------------|-----|
| Introduzione | RDR |
|--------------|-----|

TestingSummaryRep

| | |
|-----------------------|-----------------|
| Stesura del documento | RDR, VI, UA, FN |
|-----------------------|-----------------|

5.2 Dipendenze

Le dipendenze tra le attività sono rappresentate nel precedente Diagramma di Gantt e sono indicate con una linea rossa

5.3 Risorse Necessarie

E' stato stimato un insieme di tipologie di risorse necessarie alla realizzazione del progetto, di seguito classificate.

Risorse Umane

- Project Manager: Roberto Di Russo (RDR)
- Vice-Project Manager: Giovanni Maggiolini (GM)
- Staff Tecnico: Umberto Annunziata (UA) , Francesco Nicolao (FN),
Claudio Gargiulo (CG), Vincenzo Iuliano (VI)

Trattandosi di un progetto a scopi prevalentemente didattici, i ruoli previsti dal management (Architetto del sistema, Object Designer, etc) sono stati ricoperti a turno dai membri dello staff, compresi TM e VTM.

Risorse hardware

Le risorse necessarie in termini di hardware sono:

- Un computer portatile per ogni membro del team, per permettergli di svolgere i assegnati
- Una chiavetta USB per permettere lo scambio in tempo reale degli elaborati

Risorse software

Le risorse necessarie per il supporto software sono:

- OpenOffice.org Writer oppure Microsoft Word
- OpenOffice.org Impress oppure Microsoft Excel
- ArgoUML (all'inizio)
- Dia
- Microsoft Project 2007
- Eclipse
- MySQL 5.0
- Mozilla Firefox
- Sistemi Operativi Windows XP e Vista, GNU/Linux nelle distribuzioni openSUSE 10.3 e Ubuntu 7.10, Mac OS X

5.4 Allocazione del Budget e delle Risorse

Trattandosi EasyClinique di un progetto svolto a scopo didattico ed avendo esso una progettazione poco complessa, non è stata prevista alcuna strategia di allocazione di budget e di risorse.

5.5 Pianificazione

Durante lo sviluppo del prodotto SW sono previste delle date di scadenza , in gergo deadlines.

Di seguito ne viene fornito un elenco.

- 27/03/2008 - Consegna Project Proposal 1.0
- 03/04/2008 - Consegna Project Proposal 1.0
- 14/04/2008 - Consegna RAD 1.0
- 23/04/2008 - Consegna RAD 2.0
- 05/05/2008 - Consegna SDD 1.0
- 15/05/2008 - Consegna ODD 1.0
- 21/05/2008 - Consegna PMP 1.0
- 25/05/2008 - Consegna Implementazione 1.0 e Documenti di Testing 1.0
- 05/06/2008 - Consegna Documentazione finale e Testing Completo

5.6 Valutazione dello staff

E' compito del TM, supportato ove necessario dal VTM, organizzare il lavoro del team durante l'intero sviluppo.

In particolare, ogni membro dello staff viene guidato in ogni fase dal TM nello svolgere le attività indispensabili al completamento di ciascuna deliverable richiesta dal cliente nei tempi di consegna previsti, senza che ciò influisca sulla qualità dei prodotti.

Ovviamente, anche per "motivare" i componenti del gruppo, il TM provvede ad una valutazione di ciascuno di essi, secondo una serie di criteri stabiliti e resi noti. Di seguito viene riportata una lista dei parametri scelti dal TM per procedere alla suddetta valutazione.

- a. rispetto verso il lavoro degli altri componenti del gruppo
- b. rispetto dei tempi di consegna interni stabiliti
- c. qualità degli elaborati consegnati internamente al gruppo
- d. rispetto delle decisioni prese in gruppo
- e. livello di adeguamento delle convenzioni stabilite
- f. grado di atteggiamento propositivo
- g. grado di partecipazione espressa durante gli incontri
- h. capacità di lavoro di gruppo
- i. capacità di lavoro individuale
- j. percentuale di presenza agli incontri
- k. capacità di adattamento all'utilizzo di nuovi tools
- l. grado di competenze tecniche
- m. grado di entusiasmo mostrato nei confronti del sistema sviluppato

Di seguito viene riportata la scala di valori utilizzata per quantificare, per ciascun membro del team, i parametri sopra elencati.

- 0 = non sufficiente;
- 1 = sufficiente;
- 2 = buono;
- 3 = ottimo.

5.7 Meeting Agenda

Di seguito sono riportati i verbali relativi agli incontri tenuti a scopi decisionali all'inizio di ogni fase dello sviluppo

- a) [Project Proposal](#)
- b) [RAD 1.0](#)
- c) [RAD 2.0](#)
- d) [SDD 1.0](#)
- e) [ODD 1.0](#)
- f) [Implementazione](#)
- g) [Test Planning](#)
- h) [RevisioneFinale](#)