

UNIVERSITÀ DEGLI STUDI DI SALERNO



PROGETTO DI INGEGNERIA DEL SOFTWARE II
REPOMINER EVOLUTION

Software Project Management Plan

Autori:

Matteo MEROLA

Carlo BRANCA

Simone SCALABRINO

Giovanni GRANO

Supervisore:

Prof. Andrea DE LUCIA

SPMP

Versione 1.0

15 maggio 2014

Revision History

Tabella 1: Tabella delle revisioni del documento

Data	Versione	Descrizione	Autori
15/05/2014	0.9	Versione iniziale	Simone Scalabrino, Giovanni Grano, Carlo Branca, Matteo Merola
11/06/2014	0.9	Aggiunto il modello a spirale	Simone Scalabrino, Giovanni Grano, Carlo Branca, Matteo Merola

Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Evoluzione del presente documento	5
1.3	Definizioni ed acronimi	5
2	Overview del progetto	7
2.1	Obiettivi del progetto	7
2.2	Assunzioni e vincoli	7
2.3	Analisi risorse hardware e software	7
2.4	Stima dei costi del progetto	8
2.5	Deliverable del progetto	8
3	Organizzazione del progetto	10
3.1	Organizzazione strutturale	10
4	Scheduling delle attività	12
4.1	Tabella di scheduling	12
4.2	Work Breakdown Structure (WBS)	13
4.3	Diagramma di Gantt	18
5	Analisi dei rischi	19
5.1	Identificazione dei rischi	19
5.2	Strategie adottate nel risk management	20
6	Piani tecnici di processo	23

Elenco delle figure

3.1	Struttura organizzativa	10
4.1	WBS RepoMinerEvo	13
4.2	WBS Documento di analisi	14
4.3	WBS ODD	15
4.4	WBS Testing	17
4.5	Diagramma di Gantt scheduling delle attività	18
6.1	Meta-modello di sviluppo a spirale	24

Elenco delle tabelle

1	Tabella delle revisioni del documento	1
2.1	Risorse hardware/software	7
2.2	Deliverable del progetto	9
3.1	Ruoli e responsabilità	11
4.1	Scheduling delle attività	12
5.1	Identificazione dei rischi	20

1. Introduzione

1.1. Scopo del documento

Il presente documento, oltre a fornire una panoramica del progetto, descrive il processo di management che sarà portato avanti durante le fasi progettazione e sviluppo. Il documento sarà aggiornato in maniera iterativa per offrire informazioni più precise riguardo le diverse fasi del progetto.

1.2. Evoluzione del presente documento

Allo stato attuale del progetto, molte informazioni nel SPMP risultano incomplete o poco precise. Il documento sarà aggiornato e rivisto con l'avanzare dei lavori in modo da garantire una completa descrizione del processo manageriale condotto.

1.3. Definizioni ed acronimi

Definizioni:

- *Deliverables*: con il termine deliverables si ci riferisce generalmente alla documentazione tecnico/commerciale da consegnare al cliente quale risultato dell'esecuzione di una o più fasi di un progetto.
- *Scheduling*: pianificazione dei tempi e delle precedenze nell'impiego di risorse materiali ed umane per un buon svolgimento del processo di progettazione e sviluppo di un sistema software.
- *Work breakdown structure (WBS)*: rappresentazione di un progetto che consiste in una strutturazione tipicamente gerarchica delle attività che lo compongono in sotto-attività fino ad un opportuno livello di approfondimento.

- *Diagramma di Gantt*: strumento di supporto alla gestione dei progetti utilizzato principalmente nell'attività di project management quale rappresentazione dello scheduling delle attività o mansioni che costituiscono un progetto.

Acronimi:

- *ODD*: Object Design Document
- *WBS*: Work Breakdown Structure

2. Overview del progetto

2.1. Obiettivi del progetto

L'obiettivo del progetto è quello di aggiungere al framework RepoMiner un'estensione che permetta di calcolare l'entropia di un sistema software, implementando le metriche descritte nell'assignment.

2.2. Assunzioni e vincoli

RepoMinerEvo sarà realizzato come estensione del progetto RepoMiner, scritto in Java. Si assume che il progetto originario funzioni correttamente; questo sarà gestito ed evolverà indipendentemente da RepoMinerEvo.

Un vincolo, dettato dalla necessità di permettere a tutti i componenti del team di svolgere l'esame, è di completare il progetto entro il 13 luglio 2014.

2.3. Analisi risorse hardware e software

In questa sezione sono indicate le risorse hardware e software necessarie allo sviluppo del progetto. La tabella 2.1 mostra l'elenco delle risorse che si

Nome risorsa	Tipologia	Attività
Computer	Hardware	Redazione di documenti e scrittura di codice
Subversion	Software	Gestione delle versioni del codice
Draw.io	Software	Diagrammi UML
GanttProject	Software	Diagrammi Gantt
Eclipse	Software	Implementazione del codice
Editor \LaTeX	Software	Redazione documenti
Git	Software	Gestione delle versioni della documentazione

Tabella 2.1: Risorse hardware/software

prevede di utilizzare, con le relative attività a cui sono legate.

2.4. Stima dei costi del progetto

Si stima che il budget necessario per completare il progetto sia di 570 ore/uomo. Si prevede che lo sforzo sarà così ripartito:

- Analisi dei requisiti: 50h
- Impact analysis: 30h
- Object design: 90h
- Implementazione: 150h
- Testing: 220h
- Meeting: 20h
- Studio degli articoli scientifici forniti: 10h

Si provvederà ad una correzione di tali stime nei report di management.

2.5. Deliverable del progetto

I deliverable che saranno prodotti durante il progetto sono riportati nella tabella 2.2. Per RAD e SDD si farà riferimento alla documentazione esistente. La responsabilità della revisione dei deliverable è sempre di tutto il team; sono però qui specificati i componenti del team che provvederanno maggiormente alla creazione e al mantenimento di determinati artefatti. Le sigle indicano le iniziali (nome e cognome) e i numeri *1, 2 e 3* tra parentesi si riferiscono all'incremento in cui saranno creati quei deliverable.

Codice	Nome	Responsabilità	Scadenza
DL1	SPMP	Tutti	15/05/2014
DL2	Analisi Requisiti	Tutti	19/05/2014
DL3	ODD	MM e GG	9/07/2014
DL4	Test Plan	CB e SS	27/05/2014
DL5	Test Case Specification	CB e SS	1/06/2014
DL6	Codice sorgente (1)	MM e GG	31/05/2014
DL7	Test reports (1)	CB e SS	5/06/2014
DL8	Codice sorgente (2)	MM e GG	9/06/2014
DL9	Test reports (2)	MM e GG	28/06/2014
DL10	Codice sorgente (3)	CB e SS	2/07/2014
DL11	Test reports (3)	CB e SS	7/07/2014
DL12	Report di management	Tutti	13/07/2014

Tabella 2.2: Deliverable del progetto

3. Organizzazione del progetto

3.1. Organizzazione strutturale

La struttura organizzativa del progetto risulta difficile da stabilire in questa fase preliminare. In fasi più avanzate, i team member potranno essere suddivisi in team dedicati ad attività specifiche (Es. team di implementazione, team di testing).

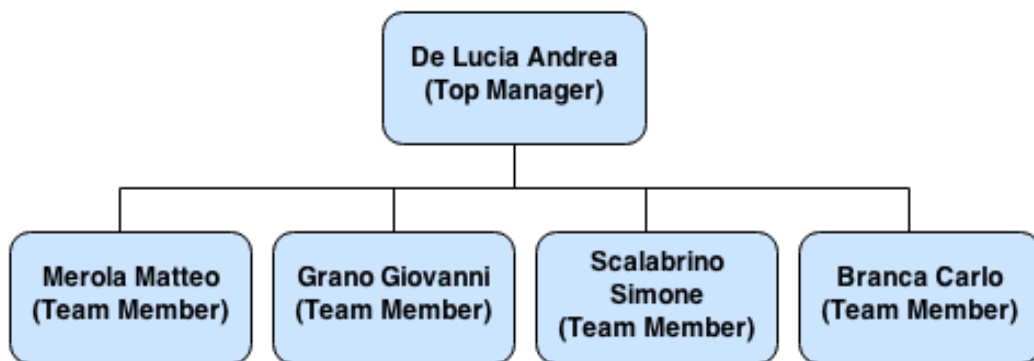


Figura 3.1: Struttura organizzativa

Nome componente	Ruolo	Responsabilità
Matteo Merola	Team member	Sviluppo deliverables: DL1, DL2, DL3, DL6, DL8, DL9, DL12
Simone Scalabrino	Team member	Sviluppo deliverables: DL1, DL2, DL4, DL5, DL7, DL10, DL11, DL12
Giovanni Grano	Team member	Sviluppo deliverables: DL1, DL2, DL3, DL6, DL8, DL9, DL12
Carlo Branca	Team member	Sviluppo deliverables: DL1, DL2, DL4, DL5, DL7, DL10, DL11, DL12

Tabella 3.1: Ruoli e responsabilità

4. Scheduling delle attività

4.1. Tabella di scheduling

La tabella 4.1 mostra lo scheduling delle macro-attività previsto.

Codice attività	Nome attività	Data inizio	Data fine	Codici deliverable
AC0	Analisi	15/05/2014	19/05/2014	DL2
AC1	Pianificazione testing	20/05/2014	1/06/2014	DL4, DL5
AC3	ODD	20/05/2014	9/07/2014	DL3
AC4	Primo incremento	20/05/2014	5/06/2014	DL6, DL7
AC5	Secondo incremento	1/06/2014	28/06/2014	DL8, DL9
AC6	Terzo incremento	23/06/2014	7/07/2014	DL10, DL11
AC7	Management	28/04/2014	13/07/2014	DL1, DL12

Tabella 4.1: Scheduling delle attività

4.2. Work Breakdown Structure (WBS)

In questa sezione è presentata la WBS del progetto e le WBS dettagliate per documento di analisi, ODD e documentazione di testing; in particolare, per le WBS della documentazione vengono presentati due formati: l'org-chart format e l'outline format.

Nell'org-chart format vengono rappresentate le componenti principali che sono dettagliate nell'outline format per questione di spazio.

4.2.1. WBS progetto

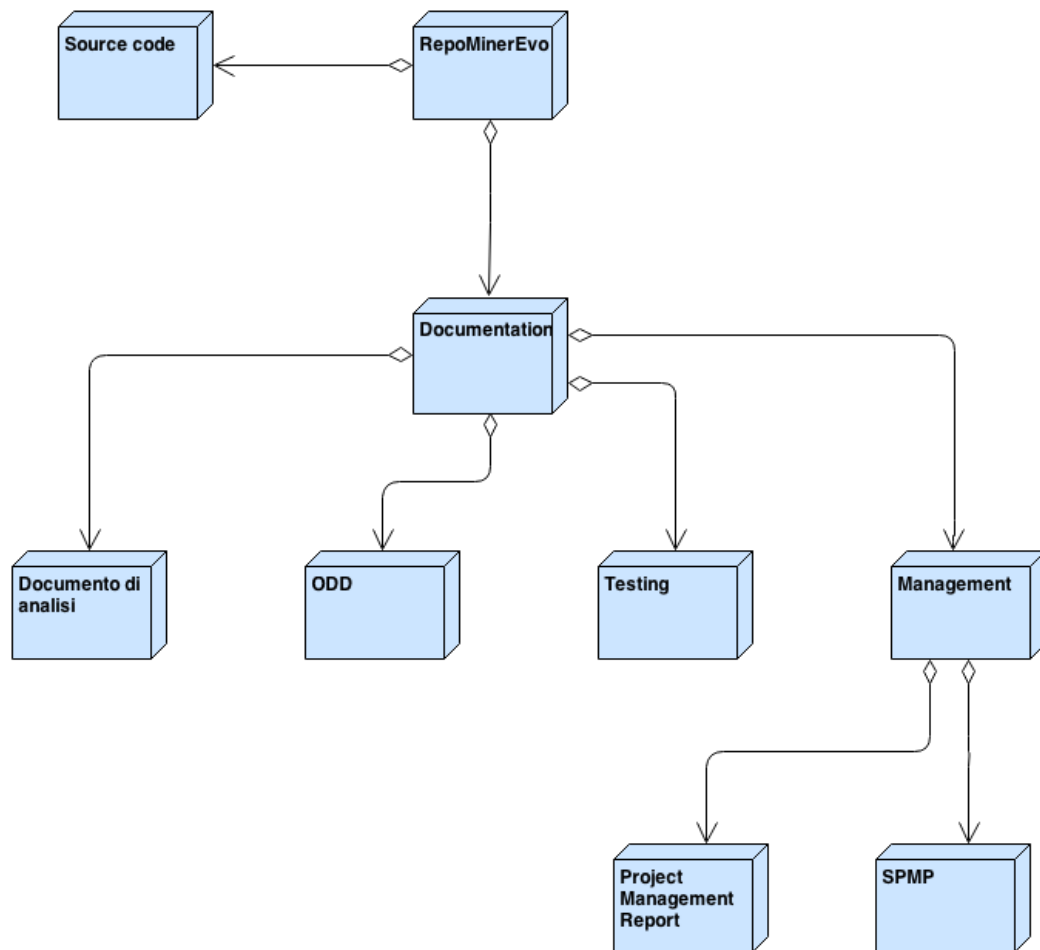


Figura 4.1: WBS RepoMinerEvo

4.2.2. WBS Documento di analisi

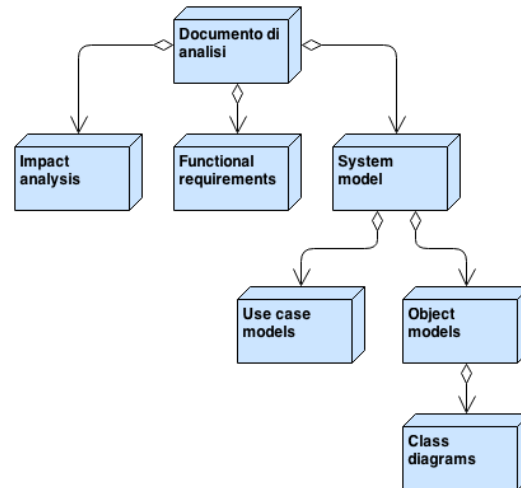


Figura 4.2: WBS Documento di analisi

WBS Documento di analisi - Outline format:

1. Introduzione
 - 1.1. Scopo del sistema
 - 1.2. Definizioni, acronimi, abbreviazioni
 - 1.3. Panoramica
 - 1.4. Sistema corrente
2. Sistema proposto
 - 2.1. Requisiti funzionali
3. System Model
 - 3.1. Use case models
 - 3.2. Object models
4. Impact analysis
5. Glossario

4.2.3. WBS ODD

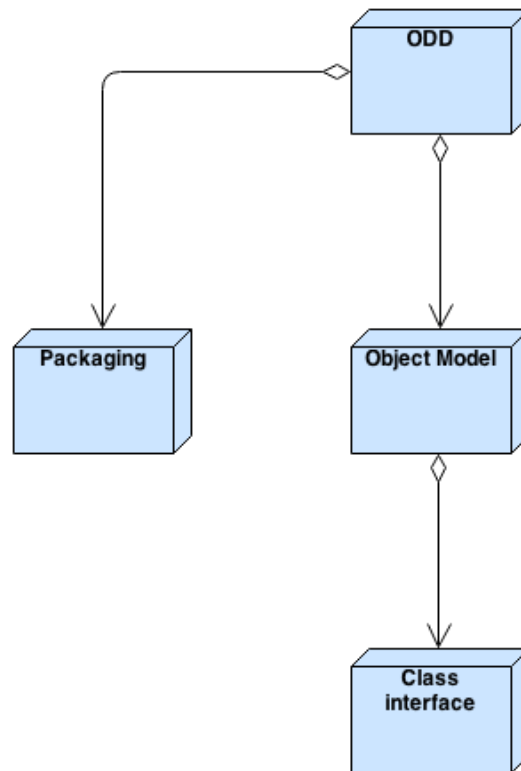


Figura 4.3: WBS ODD

WBS ODD - Outline format:

1. Introduzione
 - 1.1. Trade off dell'object design
 - 1.2. Linee guida per la documentazione delle interfacce
 - 1.3. Definizioni, acronimi e abbreviazioni
 - 1.4. References
2. Package
 - 2.1. Package
 - 2.2. Comunicazione tra i package
 - 2.3. Classi e interfacce nei package
3. Object Model

3.1. Interfaccia delle classi

4. Glossario

4.2.4. WBS testing

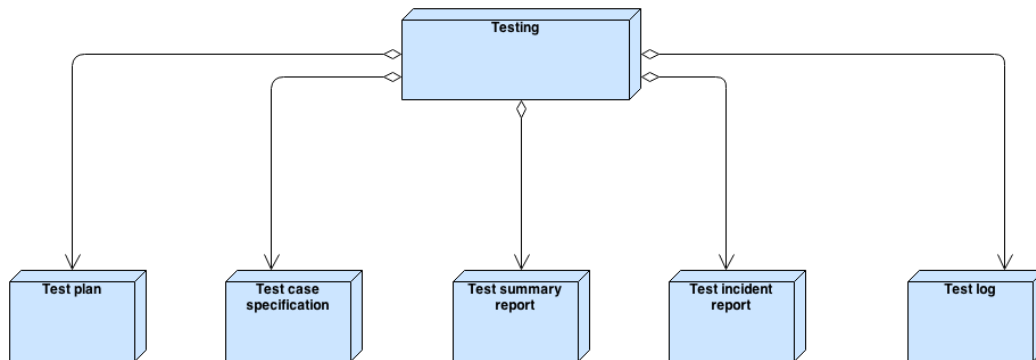


Figura 4.4: WBS Testing

WBS ODD - Outline format:

Test Plan:

1. Introduzione
2. Relazioni con altri documenti
3. Panoramica del sistema
4. Funzionalità da testare/non testare
5. Pass/Fail criteria
6. Approccio
7. Sospensione e ripristino
8. Strumenti per il testing (hardware e software)
9. Test case
10. Pianificazione del test

4.3. Diagramma di Gantt

Il diagramma di Gantt riportato in figura 4.5 rappresenta lo scheduling delle attività elencate nella sezione 4.1 del presente documento

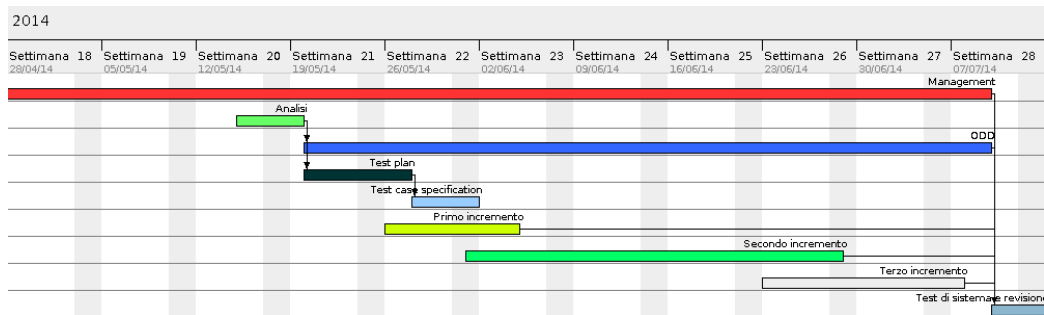


Figura 4.5: Diagramma di Gantt scheduling delle attività

Di seguito le principali scadenze:

- 19/05/14: Consegna Documento di analisi
- 27/05/14: Consegna Test Plan
- 01/06/14: Consegna Test case specification
- 04/06/14: Termine primo incremento d'implementazione
- 28/06/14: Termine secondo incremento d'implementazione
- 07/07/14: Termine terzo incremento d'implementazione
- 13/07/14: Termine fase di test di sistema e revisione

5. Analisi dei rischi

5.1. Identificazione dei rischi

In questa sezione sono identificati i possibili problemi cui il progetto potrebbe andare incontro. Per ogni rischio è indicata la probabilità attesa che si verifichi e l'impatto sulla riuscita del progetto. A seguire, nella sezione 5.2, le strategie adottate per ridurre l'impatto dei rischi sul progetto e le azioni da intraprendere nell'eventualità che un rischio si verifichi.

Legenda

Probabilità:

- *Bassa*: la probabilità che il rischio si verifichi è compresa nell'intervallo 0%-30%.
- *Media*: la probabilità che il rischio si verifichi è compresa nell'intervallo 30%-60%.
- *Alta*: la probabilità che il rischio si verifichi è compresa nell'intervallo 60%-100%.

Impatto:

- *Insignificante*: il verificarsi del rischio non compromette la buona riuscita del progetto.
- *Tollerabile*: classifica rischi di semplice gestione in quanto, le problematiche ad essi collegate, sono facilmente risolvibili.
- *Serio*: rischi di questo tipo possono rallentare notevolmente il progetto mettendone a rischio la buona riuscita; è necessario risolverli nel più breve tempo possibile.
- *Catastrofico*: rischi di difficile soluzione; se non affrontati per tempo portano di sicuro al fallimento.

Rischio	Probabilità	Impatto
Skill del team insufficienti	Bassi	Tollerabile
Poca conoscenza del dominio applicativo	Media	Serio
Perdita di un elemento del team	Bassa	Tollerabile
Ritardo consegna task	Media	Tollerabile
Implementazione non completa	Bassa	Serio
Attività prolungata oltre la scadenza prevista in fase di schedule	Media	Serio
Incontro settimanale annullato	Bassa	Tollerabile

Tabella 5.1: Identificazione dei rischi

5.2. Strategie adottate nel risk management

SKILL DEL TEAM INSUFFICIENTI

- *Prevenzione*: assegnare un task con maggior coefficiente di difficoltà ai membri più preparati. Assegnare task particolari a membri del team che hanno acquisito già in precedenza una particolare skill. Durante lo svolgimento dei meeting, cercare di fornire informazioni dettagliate sullo svolgimento dei task assegnati: definire input, lavoro da svolgere e output desiderato, fornendo eventualmente riferimenti su materiale utile per lo svolgimento del task.
- *Piano di contingenza*: privilegiare task di gruppo affiancando membri del team più preparati a coloro che mostrano maggiori lacune.

POCA CONOSCENZA DEL DOMINIO APPLICATIVO

- *Prevenzione*: prestare particolare attenzione nei meeting iniziali a discutere con il gruppo ogni aspetto del progetto, chiarendo eventuali

dubbi, sottolineandone l'utilità e il dominio applicativo in cui va a posizionarsi.

- *Piano di contingenza*: dedicare incontri straordinari alle discussioni inerenti il dominio applicativo. Domandare chiarimenti agli sviluppatori del progetto originale.

PERDITA DI UN ELEMENTO DEL TEAM

- *Prevenzione*: assegnare task complicati, quali l'implementazione, a coppie di membri del team, riducendo così il danno nel caso di verificarsi del rischio. Formare un gruppo omogeneo per quanto riguarda obiettivi e aspettative sul lavoro da svolgere.
- *Piano di contingenza*: ridistribuire i task lasciati in sospeso dal membro uscente del team tra le risorse ancora a disposizione. Considerare eventuali modifiche allo scheduling.

RITARDO NELLA CONSEGNA DEL TASK

- *Prevenzione*: fornire incentivi alle risorse nel caso di consegna puntuale del task, penalizzazioni altrimenti. In sede di meeting iniziali far notare che un ritardo su un task potrebbe causare un ritardo nella consegna del progetto collettivo, e quindi un danno per tutti i singoli elementi del team.
- *Piano di contingenza*: assegnare task più impegnativi a membri del team che dimostrano puntualità nella consegna; considerare eventuali modifiche allo scheduling.

IMPLEMENTAZIONE NON COMPLETA

- *Prevenzione*: valutare attentamente il monte ore a disposizione nello stabilire i moduli del sistema da implementare, tenendo conto anche delle esigenze dei membri del team.
- *Piano di contingenza*: documentare le difficoltà incontrare e motivare il verificarsi del rischio.

ATTIVITÀ PROLUNGATA OLTRE LA SCADENZA PREVISTA IN FASE DI SCHEDULE

- *Prevenzione*: monitorare attentamente i progressi del progetto lungo la sua durata.

- *Piano di contingenza*: riformulare lo schedule ed individuare possibili attività sulle quali recuperare il ritardo accumulato. Se il ritardo è così grave da pregiudicare la consegna finale del progetto, valutare la possibilità di procrastinarne la stessa.

INCONTRO SETTIMANALE ANNULLATO

- *Prevenzione*: stabilire gli incontri settimanalmente in accordo alle necessità e alla disponibilità di tutti i membri del team.
- *Piano di contingenza*: recuperare l'incontro o affidarsi a mezzi di comunicazione sincrona disponibili sul web.

6. Piani tecnici di processo

Per lo sviluppo del progetto si utilizzerà un Modello a Spirale: il processo di sviluppo sarà sottoposto in ogni sua fase al rispetto delle quattro task regions offerte da tale modello:

- **Determinazione di obiettivi e alternative:** obiettivi specifici per la fase del progetto
- **Valutazione delle alternative e dei rischi:** analisi dei rischi del progetto; si prendono precauzioni per ridurli o ridurne l'incidenza all'interno del progetto
- **Sviluppo e convalida:** si implementano e si testano le funzionalità previste
- **Pianificazione:** viene revisionato il progetto e si decide se effettuare un nuovo giro di spirale

La scelta è ricaduta su questo modello poiché esso rende esplicita la gestione dei rischi e permette di determinare eventuali errori in fasi iniziali.

In particolare si procederà con un'analisi generica del sistema da realizzare e, successivamente, si eseguiranno almeno due giri di spirale per implementare e testare le funzionalità incrementalmente aggiunte. Alla fine di ogni incremento il sistema deve implementare tutte le funzionalità definite per quel determinato incremento e dev'essere testato completamente.

La pianificazione del testing avverrà parallelamente rispetto al primo incremento, in modo che alla fine dell'implementazione delle prime funzionalità sia subito possibile partire con il testing e seguire il modello precedentemente descritto.

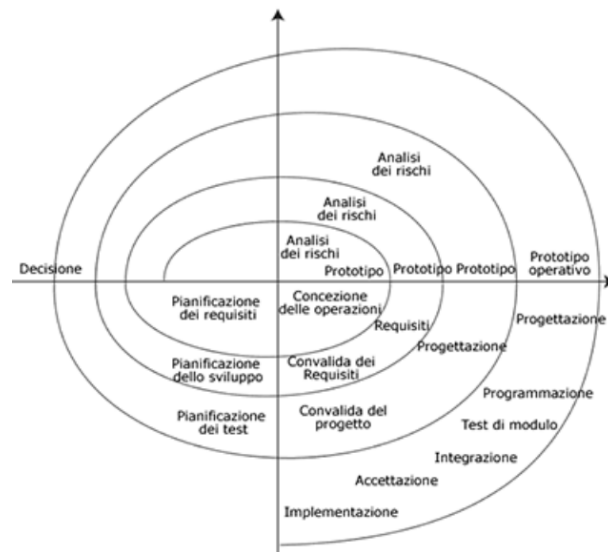


Figura 6.1: Meta-modello di sviluppo a spirale

Per la specifica delle funzionalità da implementare ad ogni incremento si rimanda all'assegno, mentre per una descrizione approfondita delle stesse si veda il documento di analisi.