

Student Notes Sharing Platform - SNSP

Project Management

Luigi Borriello

`luigi.borriello2@studio.unibo.it`

Kelvin Oluwada Milare Obuneme Olaiya

`kelvinoluwada.olaiya@studio.unibo.it`

14 agosto 2024

Indice

1	Contesto	4
2	Scoping	5
2.1	Fase Preliminare	5
2.2	Scoping Meeting	5
2.3	Conditions of Satisfaction	7
2.4	Requirements Breakdown Structure	8
2.5	SWOT Analysis	8
2.6	Project Management Lifecycle	8
2.7	Project Overview Statement - POS	8
3	Planning	9
3.1	Joint Project Planning Session	9
3.1.1	Partecipanti	9
3.1.2	Session agendas	9
3.2	Prioritizzazione dei requisiti	10
3.3	Project Definition Statement	10
3.4	Work Breakdown Structure	10
3.5	Stima delle tempistiche	11
3.6	Project Network Diagram e Gantt chart	11
3.7	Stima dei Costi	11
3.7.1	Cashflow	12
4	Launching	13
4.1	Allocazione delle risorse	13
4.2	Kick-off meeting	13
4.2.1	Agenda	13
4.3	Regole operative	14
4.3.1	Tools	14
4.3.2	Meetings	14
4.3.3	Gestione della qualità	15
4.4	Gestione cambiamenti di scope	15
5	Monitoring & Controlling	17
5.1	Sistema di Reporting	17

5.2	Issues Log	17
6	Closing	18
6.1	Approvazione da parte del committente	18
6.2	Installazione e Deployment della soluzione	18
6.3	Documentazione di progetto	18
6.4	Audit post-implementazione	19
6.5	Final Project Report	20
6.6	Chiusura del progetto	22

1 Contesto

La Student Notes Sharing Platform (SNSP) è un'iniziativa proposta dal Consiglio accademico dell'Università per migliorare la collaborazione accademica tra gli studenti. Recentemente, l'Università ha osservato un calo delle prestazioni degli studenti, attribuito alla mancanza di accesso a materiali di studio di qualità e alla collaborazione tra pari. Per risolvere questo problema, il Consiglio intende sviluppare un'applicazione software che consenta agli studenti di condividere, valutare e accedere agli appunti di studio in modo efficiente. Il dottor Gelo, Rettore dell'Università, è stato incaricato di supervisionare lo sviluppo di questa piattaforma. Il successo di questo progetto è fondamentale, in quanto potrebbe avere un impatto significativo sul rendimento e sulla soddisfazione degli studenti.

Il progetto mira a creare una piattaforma centralizzata in cui gli studenti possano caricare i loro appunti, classificarli per materia e corso e interagire con gli altri studenti attraverso commenti e valutazioni. La piattaforma deve essere intuitiva, scalabile e sicura.

2 Scoping

2.1 Fase Preliminare

Dopo aver identificato il problema si vogliono condurre sessioni di definizione dello scope. L'obiettivo finale è la formalizzazione dei requisiti (RBS e/o User stories), di un Project Overview Statement (PoS) e la scelta del Project Management Life Cycle (PMLC) oltre a delineare le Conditions of Satisfaction (CoS).

2.2 Scoping Meeting

Di seguito viene riportata una descrizione delle riunioni tenute per la definizione dello scope di progetto.

- **Primo incontro:** Dopo la presentazione dei membri del progetto (membri del reparto IT, project manager e gli stakeholders lato committente) il rettore ha presentato la problematica rilevata nella sua Università. Successivamente si è iniziato a discutere sulle opportunità da cogliere al fine di risolvere la problematica presentata. Si è convenuto sull'idea di realizzare una piattaforma per lo scambio di appunti. Inoltre, sono state delineate le Condition of Satisfaction (CoS) e valutate eventuali soluzioni Off-The-Shelf (OTS). A conclusione del meeting è stata definita l'agenda del prossimo incontro, che include la definizione dei dettagli dell'applicazione da realizzare.

Hanno partecipato:

- Il project manager
- Il rettore
- Membri del reparto IT
- Un facilitatore

Deliverables prodotti:

- CoS - Condition of Satisfaction
- **Secondo incontro:** Nel secondo incontro, come da ordine del giorno (OdG),

sono stati discussi i dettagli dell'applicazione e identificati i requisiti principali del sistema. In seguito a ciò, si è convenuto sulla scelta di condurre il progetto come un insieme di più sottosistemi che, molto verosimilmente saranno micro-servizi (si è deciso di rettificare questo aspetto in fase di planning). Quindi, per ciascuno di essi, sono state definite le CoS. Infine come programma della terza sessione è stato dato come obiettivo la finalizzazione della stesura dei requisiti dei vari sottosistemi e la scelta del Project Management Life Cycle (PMLC).

Hanno partecipato:

- Il project manager
- Il rettore (Dr. Gelo)
- Membri del reparto IT
- Un facilitatore

Deliverables prodotti:

- Raffinamento delle CoS.
- **Terzo incontro:** Il terzo incontro è stato dedicato principalmente alla stesura formale dei requisiti tramite interviste ad alcuni end-user, per ciascun sottosistema.

Hanno partecipato:

- Il project manager
- Membri del reparto IT
- Un facilitatore
- Un gruppo di studenti dell'Università (end-users)

Deliverables prodotti:

- Prima bozza delle RBS
- **Quarto incontro:** Nel quarto incontro è stato presentato un prototipo dell'applicativo agli studenti presenti alla riunione, e una volta approvato, è stata condotta un'analisi SWOT per ciascun sottosistema. In seguito a ciò il team

di sviluppo ha proposto i vari PMLC.

Hanno partecipato:

- Il project manager
- Membri del reparto IT
- Un facilitatore
- Un gruppo di studenti dell'Università (end-users)

Deliverables prodotti:

- SWOT Analysis
 - Raffinamento della RBS.
- **Quinto incontro:** Nel quarto incontro, dopo l'approvazione dei requisiti da parte del rettore, è stata finalizzata la definizione del POS.

Hanno partecipato:

- Il project manager
- Il Rettore (Dr. Gelo)
- Membri del reparto IT
- Un facilitatore

Deliverables prodotti:

- POS - Project Overview Statement.

2.3 Conditions of Satisfaction

Come prima cosa, son state definite le Conditions of Satisfaction (CoS), disponibili in allegato.

2.4 Requirements Breakdown Structure

Sebbene il progetto sia stato commissionato dal rettore dell'università, il risultato del progetto stesso sarà destinato principalmente all'uso da parte degli studenti. Ci si aspetta che il rettore non abbia una piena consapevolezza della vita dello studente universitario, pertanto, è stato favorito un approccio a interviste e prototyping per la raccolta dei requisiti direttamente con gli studenti, con la speranza che possano rispondere al meglio alle esigenze di quest'ultimi.

In allegato è possibile trovare gli RBS formalizzati per ciascun progetto.

2.5 SWOT Analysis

Le SWOT analysis per ciascun sottosistema del progetto SNSP forniscono una visione dettagliata dei punti di forza, debolezze, opportunità e minacce che potrebbero influenzare lo sviluppo e l'implementazione della piattaforma. Queste analisi aiutano a identificare le aree che richiedono maggiore attenzione e le strategie per sfruttare al meglio le opportunità e mitigare i rischi.

2.6 Project Management Lifecycle

Data la completezza e la chiarezza dei requisiti, abbiamo optato per un approccio di tipo Tradizionale Lineare.

2.7 Project Overview Statement - POS

Il risultato ottenuto dall'ultima sessione di scoping meeting è il POS, disponibile in allegato, il quale descrive, con una visione ad alto livello, i dettagli del progetto.

3 Planning

Anche nel planning lo strumento fondamentale sono le riunioni. Durante il Joint Project Planning Session (JPPS) si definisce cosa fare per soddisfare i requisiti entro i vincoli di tempo, di budget e di risorse disponibili.

3.1 Joint Project Planning Session

Per il planning sono state necessarie due sessioni JPPS.

3.1.1 Partecipanti

Di seguito, un elenco dei partecipanti alle sessioni:

- Un facilitatore di sessione
- Il Project Manager
- Membri del personale IT
- Un rappresentante del committente
- Tecnografo

3.1.2 Session agendas

Prima sessione

- Kick-off del planning
 - Introduzione del committente
 - Panoramica del progetto, e inquadramento dello stesso nella strategia del committente;
 - Introduzione del project manager;
 - Introduzione del personale IT;
- Working session

- Validazione e prioritizzazione dei requisiti;
- Generazione del PDS;

La prima sessione è stata l'occasione per introdurre ai presenti il progetto

Seconda Sessione

- Working session
 - Generazione e validazione della WBS;
 - Stima della durata delle attività di progetto e delle risorse richieste;
 - Costruzione del Project Network Diagram;

3.2 Prioritizzazione dei requisiti

Le considerazioni fatte riguardo la prioritizzazione dei requisiti hanno portato alla conclusione che, essendo l'applicazione composta principalmente da feature di moderata complessità, tutti i requisiti siano da classificarsi, secondo l'approccio *MoSCoW*, come requisiti “**Must** have”. Solo alcuni dei requisiti sono stati classificati come “**Should** have”. Quest'ultimi sono principalmente requisiti non funzioni legati ad aspetti di performance. È possibile visualizzare quest'ultimi, nell'allegato relativo alle RBS, colorati di [blu](#).

3.3 Project Definition Statement

A partire dal POS, è stato realizzato il *Project Definition Statement*. Il documento ha dato la possibilità di chiarire e formalizzare i vincoli in termini di tempo, budget e risorse per il progetto. È possibile trovare il documento in allegato.

3.4 Work Breakdown Structure

Nella seconda sessione di JPPS, a partire dai vari RBS, sono stati redatti i vari WBS. Questi sono disponibili in allegato.

3.5 Stima delle tempistiche

Per stimare i tempi del progetto, abbiamo optato per l'utilizzo di un approccio "consensus-based". In particolare abbiamo utilizzato la tecnica Delphi. Le durate delle attività sono state stimate in *ore/uomo*.

Le stime delle attività sono disponibili in allegato.

3.6 Project Network Diagram e Gantt chart

Per determinare la schedula delle attività di progetto abbiamo costruito dapprima il *Project Network Diagram* nel quale abbiamo riportato le dipendenze temporali fra le varie attività. Successivamente abbiamo derivato e rifinito il *Gantt Chart*.

Sia il PDN che il Gantt sono disponibili come allegati.

3.7 Stima dei Costi

Per la realizzazione dei vari microservizi sono stati preventivati i seguenti team:

- Un team di 2 sviluppatori per il microservizio "Notes Repository".
- Un team composto da 2 sviluppatore e 2 data scientist per il microservizio "Recommendation Engine".
- Un team di 2 sviluppatori per il microservizio "User Authentication and Profile Management".
- Un team composto da 2 UI/UX Designer e 2 sviluppatori per il Frontend.

Per la stima dei costi si ipotizza una retribuzione compresa tra €20 e €35 per ora / uomo, in base al contratto in essere. Si prevede inoltre una giornata lavorativa di 8 ore. Hardware, costi di manutenzione e consumi saranno inclusi nel costo finale del progetto.

3.7.1 Cashflow

Dato che per gran parte del team dedicato alla realizzazione del microservizio di Raccomandazione questo progetto rappresenta la prima occasione in cui adottare tecniche di machine learning ai fini della raccomandazione personalizzata dei contenuti, per non correre il rischio di trovarsi con il budget esaurito, si è ritenuto opportuno optare per un contratto a consuntivo. Nella redazione del cash flow è stato dunque tenuto in considerazione:

- Un anticipo.
- Gli stipendi mensili del team di sviluppo.
- I costi relativi all'infrastruttura necessaria all'hosting dei servizi e soprattutto al training dei modelli di machine learning.
- I costi delle licenze.
- I costi dovuti dall'uso dei servizi di terze parti.
- Gli introiti relativi al rilascio dei risultati intermedi.

4 Launching

4.1 Allocazione delle risorse

Come previsto in pianificazione, per ogni microservizio viene allocato un team composto da due sviluppatori di cui uno senior e uno junior. Per il solo microservizio di raccomandazione, al core team vengono affiancati due data scientist.

Per il frontend sono stati assunti a contratto due UI/UX designer.

È inoltre presente un project manager. Gli sviluppatori senior di ciascun core team assumeranno anche il ruolo di co-project manager.

Il committente si impegna a formare un gruppo di persone che saranno funzionali a guidare e valutare i risultati intermedi prodotti dal team di sviluppo. Questi saranno principalmente studenti che prenderanno parte alle varie sessioni di Focus Group, experience prototyping, etc.

4.2 Kick-off meeting

Il kick-off meeting ha rappresentato un momento fondamentale per il progetto, sancendo l'inizio della fase esecutiva. Durante l'incontro, è stata seguita un'agenda dettagliata che ha fornito al team di sviluppo una chiara roadmap delle attività da portare a termine.

4.2.1 Agenda

- Introduzione
- Presentazione del client team.
- Presentazione del progetto e degli obiettivi.
- Presentazione della pianificazione.
- Discussione delle regole operative per il team.
- Sessione di domande e risposte.

4.3 Regole operative

Per assicurare un'efficace gestione delle attività di progetto, abbiamo definito le seguenti regole operative.

4.3.1 Tools

Task management Al fine di ottimizzare la gestione delle attività, si è deciso di adottare l'applicativo *Trello*. Il tool permetterà al team di beneficiare di una maggiore visibilità sullo stato delle attività. Inoltre consentirà una migliore organizzazione del lavoro e una comunicazione più efficace.

Version control Per il versionamento del codice sorgente verrà adottato *Git*. Ogni microservizio verrà gestito come modulo separato. Ciò consentirà di isolare le modifiche limitando i conflitti nelle operazioni di integrazione di nuove funzionalità. Per promuovere la collaborazione tra i membri del team il workflow di sviluppo di baserà intorno alle *pull requests*. Ciò garantirà una revisione accurata del codice prima che questo venga integrato nel ramo di sviluppo principale.

Comunicazioni Le comunicazioni avverranno principalmente attraverso la piattaforma *Slack*. Verranno definiti dei canali appositi per ciascun topic inerente al progetto. Slack sarà inoltre lo strumento attraverso il quale verranno condotti i diversi meeting. Verrà sfruttata anche l'integrazione con l'applicativo *Trello* in modo che tutti i membri del team siano al corrente dello stato di avanzamento del progetto.

4.3.2 Meetings

Per ogni giornata lavorativa è previsto un *Daily Status Meeting* di 20 minuti per pianificare il lavoro.

In caso di problematiche riscontrate durante lo sviluppo, sono previste dei *Problem Resolution Meeting*.

Nell'eventualità in cui sarà necessario prendere decisioni che potrebbero avere impatto sul progetto, come ad esempio i cambiamenti di scope, sarà possibile condurre dei *Decision Making Meeting* per gestire le problematiche del caso.

4.3.3 Gestione della qualità

In termini di qualità è stato concordato che:

- tutte le attività inenti al progetto che possono essere automatizzate dovranno esserlo utilizzando un apposito build tool.
- Verranno utilizzati i cosiddetti *Git Hooks*. Lo scopo sarà quello di controllare che ogni *commit* sia relativo a codice ben formattato, privi di bug più comuni e che il codice sia correttamente documentato.
- Una volta che i commit verranno pubblicati sul repository remoto, verrà eseguita su di essi una pipeline di *Continuous Integration*. Per fare ciò si farà uso delle *GitHub Actions*. Le pipeline si occuperanno di eseguire una *build* dei vari microservizi, eseguire tutti i test automatizzati e generare eventuali resoconti.
- I messaggi di commit dovranno seguire le indicazioni di conventional commit. In questo modo sarà possibile sia tenere traccia dello storico delle modifiche ma anche automatizzare il processo di rilascio e versionamento. Infatti, messaggi di commit opportunamente costruiti innescheranno il *deploy* del sistema e della documentazione in apposite pagine statiche.

4.4 Gestione cambiamenti di scope

Il processo di gestione dei cambiamenti di scope prevede una sequenza di fasi ben definite:

- Formalizzazione della richiesta: le richieste dovranno pervenire al Project Manager via Slack per una prima valutazione.
- Analisi preliminare: il PM valuta l'impatto della modifica e la presenta al team nel primo daily meeting disponibile.
- Decision Making Meeting: se necessario, si organizza un meeting formale per prendere una decisione definitiva.
- Approvazione del cliente: si coinvolge il cliente per ottenere una sua approvazione.

- Aggiornamento del piano: qualora il cambiamento venga approvato, PM aggiorna il piano di progetto e comunica le modifiche al team.

5 Monitoring & Controlling

5.1 Sistema di Reporting

Per garantire un costante monitoraggio dell'avanzamento del progetto, saranno previste riunioni settimanali dedicate all'aggiornamento dello stato delle attività. Come deliverable del meeting ogni team dedicato a un microservizio produce un report di tipo *Stoplight*.

In particolare:

- il **verde** starà a indicare che le attività stanno venendo completate nei tempi stabiliti.
- il **giallo** starà a indicare che ci sono attività in ritardo o con potenziali rischi.
- il **rosso** indicherà che ci sono attività con forte criticità.

Questa visualizzazione consentirà di intervenire tempestivamente sulle eventuali deviazioni dal piano.

5.2 Issues Log

La piattaforma GitHub, oltre a essere un ottimo strumento di versionamento, offre una serie di strumenti avanzati tra cui un sistema per la segnalazione e il tracciamento delle **issues**. Tramite quest'ultimo sarà anche possibile etichettare le issues aperte e fornire commenti dettagliati. Un'ulteriore importante funzionalità disponibile è la chiusura automatica delle issue una volta che queste vengono risolte. Ciò permetterà di mantenere un registro preciso e puntuale sullo stato del progetto.

6 Closing

6.1 Approvazione da parte del committente

Il processo di approvazione del cliente ha previsto in prima battuta la certificazione che tutti i test, precedentemente concordati, eseguono con successo. Inoltre, è stata messa in piedi un'infrastruttura di collaudo. Quest'ultima ha permesso al cliente di testare tutte le funzionalità del sistema, in particolare quelle relative raccomandazione. Una volta ricevuto l'approvazione del cliente si è proceduto con le operazioni di installazione.

6.2 Installazione e Deployment della soluzione

Non disponendo di un sistema già in funzione da sostituire si è optato per un approccio “*simil-Cut-Over*”.

6.3 Documentazione di progetto

Fin dalla fase di sviluppo, abbiamo curato con grande attenzione la creazione di una documentazione completa e dettagliata, coprendo ogni aspetto del progetto, dalla fase di progettazione a quella di implementazione. Questa documentazione serve come riferimento fondamentale per tutti gli stakeholder e faciliterà le attività di manutenzione e aggiornamento future.

Nello specifico, la documentazione è così composta:

- breve guida utente, con le istruzioni per registrarsi, visualizzare, caricare, scaricare e valutare note nella piattaforma.
- una guida per gli sviluppatori con le linee guida e regole sulle convenzioni da utilizzare per i commit e per la gestione dei rami di sviluppo.
- report tecnico dove vengono illustrate le caratteristiche architetturali del sistema e le tecnologie utilizzate.
- report sugli *Acceptance Tests* effettuati, come attestato del soddisfacimento delle specifiche richieste.

6.4 Audit post-implementazione

Al fine di garantire il miglioramento continuo, abbiamo condotto un audit post-implementazione dopo aver ricevuto il via libera dagli stakeholder. Questo processo ci ha permesso di valutare il successo del progetto e di identificare le azioni necessarie per ottimizzare i risultati futuri.

Per fare ciò, abbiamo riflettuto sulle seguenti domande:

- *Gli obiettivi del progetto sono stati raggiunti?* Sì, il corretto funzionamento del sistema attesta il raggiungimento degli obiettivi prefissati.
- *Il deliverable fa quello che si aspettava il committente?* Sì, gli acceptance-test definiti sui requisiti funzionali, attestano l'aderenza alle specifiche richieste.
- *Il progetto è stato completato rispettando i limiti di tempo, budget e rispettando le specifiche?* Sì, il progetto è stato completato rispettando i limiti di tempo, budget e specifiche. Le attività di monitoraggio sono state di vitale importanza in quanto hanno consentito di attuare tempestivamente manovre correttive.
- *Il committente è soddisfatto del risultato del progetto?* Sì, il committente ha espresso chiara soddisfazione nei confronti del risultato ottenuto.
- *Il business value previsto si è concretizzato?* La fase di collaudo ha fornito risultati promettenti vista la chiara soddisfazione del committente. Tuttavia, un'analisi obiettiva sul raggiungimento del business value previsto sarà possibile solo dopo almeno 3-6 mesi di utilizzo del sistema da parte del committente.
- *I criteri di successo sono stati rispettati?* Il progetto ha rispettato pienamente i criteri di successo definiti in fase di pianificazione con ottimi standard di performance e qualità.
- *Che lezione è stata imparata relativamente alla metodologia di gestione del progetto scelta?* L'approccio lineare ha garantito che ogni fase del progetto fosse completata prima di passare alla successiva, riducendo il rischio di errori imprevisti. Tuttavia, questa rigidità ha limitato la capacità di adattarsi ai cambiamenti in corso d'opera. In alcuni casi, questo ha portato a lievi ritardi o alla necessità di rivedere piani già approvati. In futuro potremmo integrare una maggiore flessibilità, ad esempio adottando un approccio *Agile*,

per poter rispondere meglio a nuove esigenze senza il rischio di compromettere l'esecuzione del progetto.

- *Come ha seguito la metodologia il team?* Il team ha seguito bene la metodologia, grazie anche alla documentazione a loro fornita. Inoltre ha apprezzato molto i *Daily status Meeting* e le riunioni settimanali. Hanno permesso di far emergere tempestivamente tutte le questioni relative al progetto e dunque di tenere basso il livello di stress. Questo conferma quanto detto prima, ovvero, che in futuro, l'uso di un approccio Agile (che include momenti di aggiornamento periodico e di retrospettiva) potrebbe essere più adatto a progetti di questo tipo.

6.5 Final Project Report

Executive Summary L'idea del progetto era di realizzare una piattaforma che permettesse a studenti universitari di scambiare appunti e note relativi ai diversi corsi offerti dall'università.

La realizzazione del progetto è andata a buon fine e ha rispettato il budget e i tempi previsti.

La chiarezza dei requisiti e degli obiettivi, ha fatto sì che non si siano presentati gravi imprevisti, sebbene sia stato adottato un approccio abbastanza "rigido" come quello lineare.

Livello di successo e performance complessive del progetto Il progetto è stato portato a termine con successo, consegnando un sistema software che soddisfa pienamente le esigenze dei nostri clienti. Tutte le funzionalità richieste sono state implementate con successo e il prodotto finale ha superato tutti i test di qualità. La piattaforma messa a disposizione degli studenti rappresenta un nuovo strumento di collaborazione e condivisione delle conoscenze. Si prevede che possa contribuire a migliorare i risultati accademici, anche se saranno necessari ulteriori dati per confermare questa ipotesi.

Organizzazione e amministrazione del progetto La gestione del progetto è stata guidata dal Project Manager, che ha coordinato le attività dei vari team durante

tutto il ciclo di vita del progetto.

Una volta identificati i microservizi in fase di design, ciascuno di essi è stato affidato a un singolo Team. I processi di design e sviluppo dei vari microservizi sono stati quindi parallelizzati il più possibile.

Pregi dell'approccio utilizzato (sia dal punto di vista architetturale che di gestione del progetto)

- **Struttura chiara e definita:** Ogni fase del progetto (analisi, progettazione, sviluppo, test, implementazione) è ben delimitata e segue una sequenza logica.
- **Facilità di gestione:** La struttura lineare rende più semplice la pianificazione, il controllo e la gestione del progetto.
- **Documentazione completa:** Ogni fase è documentata in modo dettagliato, facilitando la comprensione del progetto e la manutenzione successiva.
- **Adatto a progetti stabili:** Funziona bene per progetti con requisiti ben definiti e stabili nel tempo, dove non sono previste molte modifiche durante lo sviluppo.

Difetti dell'approccio utilizzato

- **Rigidità:** Una volta iniziata una fase, è difficile tornare indietro e apportare modifiche alle fasi precedenti.
- **Scarsa flessibilità:** Non è adatto a progetti complessi o in continua evoluzione, dove i requisiti possono cambiare frequentemente.
- **Rischio di consegnare un prodotto non rispondente alle esigenze dell'utente:** Il cliente vede il prodotto finito solo alla fine del progetto, aumentando il rischio di insoddisfazione.
- **Tempi di consegna lunghi:** A causa della sequenzialità delle fasi, i tempi di sviluppo possono essere lunghi.
- **Difficoltà nel gestire l'incertezza:** L'approccio lineare presuppone una conoscenza completa dei requisiti iniziali, il che può essere difficile da garantire in progetti complessi.

6.6 Chiusura del progetto

Attraverso la firma del rapporto finale, gli stakeholder confermano la conclusione formale del progetto e riconoscono il valore del lavoro svolto. Possiamo dunque ritenere il progetto concluso.