



Piano di progetto



IO, LEGO

Autori

- Volpe Davide 862989
- Melissari Luca 865700
- Jasari Besar 862699
- Marcolin Giulio 862611
- Bertin Eleonora 865370

Sommario

1. Introduzione	2
1.1 Overview del progetto	2
1.2 Deliverables del progetto	2
1.3 Evoluzione del progetto	2
1.4 Materiale di riferimento	3
1.5 Definizioni ed abbreviazioni	3
2. Organizzazione del progetto	4
2.1 Modello del progetto	4
2.2 Struttura organizzativa	4
2.3 Interfacce organizzative	4
2.4 Responsabilità del progetto	5
3. Processi gestionali	5
3.1 Obiettivi e priorità	5
3.2 Assunzioni, dipendenze, vincoli	5
3.3 Gestione dei rischi	6
3.4 Meccanismo di monitoraggio e controllo	8
3.5 Pianificazione dello staff	8
4. Processi tecnici	8
4.1 Metodi, strumenti, tecniche	8
4.2 Documentazione del software	9
4.3 Funzionalità di supporto al progetto	9
5. Pianificazione del lavoro, delle risorse umane e del budget	10
5.1 WBS (Word Breakdown Structure)	10
5.2 Dipendenze	12
5.3 Risorse necessarie	12
5.4 Allocazione del budget e delle risorse	12
5.5 Pianificazione	13

1. Introduzione

1.1 Overview del progetto

Il progetto consiste nella creazione di una applicazione Android pensata per comandare un robot Lego attraverso apposite API.

Essa si presenta come un'interfaccia mobile con appositi tasti e comandi e sarà user-friendly, appositamente creata per supportare eventuali modifiche all'applicazione o al robot. Il progetto sarà quindi elastico e adattabile ai cambiamenti nel tempo. Sarà, inoltre, interattivo in quanto verranno usati diversi sensori per permetterne funzioni più complesse e utili.

1.2 Deliverables del progetto

Le scadenze per la consegna della documentazione riguardante il progetto sono le seguenti:

Oggetto	Deadline
• Piano di Progetto:	16/10/2018
• Documento di analisi e specifica:	02/11/2018
• Piano di testing:	15/11/2018
• Documento di Progettazione:	10/12/2018
• Sviluppo e Delivery:	31/01/2019

1.3 Evoluzione del progetto

Il progetto vedrà la seguente evoluzione:

- Stesura della documentazione
- Analisi delle richieste, ideazione dell'applicazione e ricerca dei dati
- Progettazione dell'applicazione
- Sviluppo del codice

Durante il corso dello sviluppo del progetto è possibile che vengano apportate modifiche alle funzionalità dell'applicazione senza però andare a modificare la struttura del progetto e le funzionalità già previste dell'applicazione.

1.4 Materiale di riferimento

I materiali di consultazione durante la progettazione sono :

- Slides del corso di Ingegneria del Software
- Android Programming For Beginners - Jhon Horton
- Android 5 Programming by Example - Kyle Mew
- Guida di android studio ufficiale

<https://developer.android.com/studio/intro>

1.5 Definizioni ed abbreviazioni

Applicazione:

Software il cui scopo è rendere possibile una o più funzionalità, servizi o strumenti all'utente.

Android:

Sistema operativo sviluppato da Google, basato su kernel Linux.

Feedback:

Giudizio inviato da un utente allo sviluppatore contenente eventuali segnalazioni di errori.

Bug:

Errore o guasto che porta al malfunzionamento del software. Generalmente si tratta di scrittura di codice non corretto da parte del programmatore.

Versione Beta:

Versione di software non definitiva, ma già parzialmente testata. Gli utenti scelti per i test della versione Beta tentano di scoprire nuovi bugs tramite azioni non previste e li riportano tramite feedback.

GitHub:

Servizio di hosting per progetti software. Permette, tramite il software GIT, il controllo versione del progetto.

2. Organizzazione del progetto

2.1 Modello del progetto

Il modello che andremo ad utilizzare sarà evolutivo. Studieremo ed analizzeremo la fattibilità della realizzazione del nostro programma determinando le modalità di svolgimento di tutte le fasi.

Come detto in precedenza, però, prevediamo che questo progetto comporti diverse modifiche in corso d'opera, in quanto la scelta e l'utilizzo dei sensori del robot può portare alla modifica di determinati utilizzi e compiti decisi in precedenza per eventuali dati non reperibili o inattese funzionalità. Perciò, proveremo e testeremo diverse volte le funzionalità della nostra applicazione e una volta raggiunto un punto dove il prototipo funziona correttamente, risulta essere coerente con la descrizione nel nostro piano e gli eventuali bug non porteranno alla rottura dei componenti o al perenne malfunzionamento, metteremo in commercio la nostra applicazione e successivamente faremo gli eventuali aggiornamenti per renderlo il più perfetto possibile.

2.2 Struttura organizzativa

Il gruppo del progetto avrà una struttura democratica dove ogni componente nel gruppo avrà gli stessi poteri decisionali.

Nel caso un componente abbia una proposta per la modifica o l'aggiunta di qualche funzionalità al progetto essa verrà prima discussa con altri membri del gruppo.

2.3 Interfacce organizzative

Telegram:

Attraverso un un gruppo comprendente tutti i membri del progetto possiamo discuterne e organizzare eventuali attività.

Github:

Creazione di una repository privata dove condividere il progetto con tutti i membri del gruppo, in modo tale da permettere ad ognuno a svolgere il suo compito.

Bacino di utenti:

Per testare la nostra applicazione ci affideremo a un bacino di utenti da cui avere un feedback.

Prof. Agostino Cortesi e Prof. Spanò Alvise:

Durante lo sviluppo avremo degli incontri con i professori per avere una valutazione e un feedback sul progetto.

2.4 Responsabilità del progetto

La seguente lista contiene una visione dei compiti da svolgere durante la progettazione dell'applicazione. Questi, però, non saranno affidati rigorosamente ad un'unica persona in quanto ogni membro può dare il suo contributo al lavoro degli altri in modo da evitare eventuali ritardi o perdite di lavoro.

Non viene quindi attribuito nessun compito a specifiche persone.

- Stesura documentazione
- Montaggio del robot
- Sviluppo app, parte interfaccia utente
- Sviluppo app, parte interazioni con componenti del robot
- Test e correzione dei bug

3. Processi gestionali

3.1 Obiettivi e priorità

Dato che il modello del progetto è di tipo evolutivo la priorità del gruppo sarà di sviluppare una beta dell'applicazione per android e un prototipo funzionante del robot, cosicché si potrà avere un feedback su di essi e di conseguenza passare alla fase di miglioramento ed eventuali modifiche fino alla versione finale.

L'obiettivo finale è di produrre un robot e una applicazione android efficiente e ben sviluppata.

3.2 Assunzioni, dipendenze, vincoli

Assumiamo che:

- Il cliente abbia un dispositivo Android e sappia attivare il Wi-Fi per il collegamento con il robot
- Il cliente abbia un'età minima di 10 anni per l'utilizzo del robot Lego Mindstorm
- L'utente non deve compromettere il corretto funzionamento del Robot, alterando la sua struttura o danneggiando alcune componenti.

Dipendenze:

- Il software verrà sviluppato nell'ambiente Android Studio in linguaggio Java.
- Il dispositivo Android dell'utente deve avere versione minima Android 5

Vincoli:

- Rispetto dei vincoli temporali definiti dalle scadenze delle varie consegne definite nel punto 1.2

3.3 Gestione dei rischi

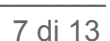
- PS = Product size
- ST = Staff size and experience
- DE = Development environment

Rischi	Categoria	Probabilità	Impatto	RMM
Problemi di saluti di qualche membro	ST	60%	Basso	#1
Membri del team che non lavorano bene insieme	ST	30%	Medio	#2
Componenti del robot non funzionanti	DE	20%	Medio	#3
Sovrapposizione di attività universitarie / lavorative differenti per i membri del gruppo che causano rallentamenti allo sviluppo del progetto	DE	50%	Medio	#4
Parti di codice separatamente sviluppate non corrette o non compatibili richiedono una ri-stesura	DE	60%	Medio	#5
Sviluppo di funzionalità previste non più compatibili con il robot per problemi di natura tecnica	DE	10%	Basso	#6
Sviluppo di parti del progetto richiedono più tempo di testing, design, implementazione e montaggio di quanto previsto.	DE	80%	Medio	#7
Problematiche derivanti dal lavoro con strumenti e ambienti di sviluppo non familiari	DE	20%	Basso	#8
Perdita di membri durante lo sviluppo del progetto	ST	5%	Medio	#9
Cattiva stima della taglia del progetto	PS	70%	Alto	#10

Soluzioni ai diversi RMM:

- 1) Gli altri componenti del gruppo tentano di svolgere in parte il lavoro spettante al membro indisponibile.
- 2) Si cerca, tramite il dialogo, di risolvere i conflitti interni e di creare un ambiente di lavoro pacifico per tutti.
- 3) Si richiede immediata sostituzione da parte del Professore e se non disponibile, si cerca una componente uguale o simile.

- ## RISK



3.4 Meccanismo di monitoraggio e controllo

I controlli sulla qualità del lavoro e sulla correttezza del codice di progetto saranno divisi in due categorie:

- Controlli pianificati: almeno uno/due controlli generali a settimana. Controlli aggiuntivi in caso di rilevazione di problemi durante l'implementazione del codice
- Controlli individuali: Ogni membro avrà diritto di revisione e modifica se e quando sarà ritenuto necessario, informando gli altri membri del cambiamento eseguito

I principali canali di comunicazione e condivisione saranno Telegram e Discord.

Per evitare la perdita di dati verranno utilizzati strumenti quali Drive e GitHub.

3.5 Pianificazione dello staff

Per lo sviluppo del progetto sono necessarie alcune competenze tra le quali:

- La conoscenza della programmazione ad oggetti in particolare Java
- La conoscenza di xml
- Saper utilizzare Android Studio
- Saper usare GitHub

Qualora mancasse una o più di queste competenze a uno dei membri del gruppo esso è tenuto ad impararle.

4. Processi tecnici

4.1 Metodi, strumenti, tecniche

Per sviluppare il progetto verranno usate le seguenti risorse:

- Software Android Studio
- Repository privata nella piattaforma GitHub
- 5 computer portatili personali. Il sistema operativo utilizzato non è significativo, poiché lo sviluppo in ambiente Android Studio è totalmente indipendente dal sistema operativo.
- Dispositivi Android personali
- Lego Mindstorms ev3 in dotazione
- Google Documents per la redazione dei documenti

Metodi

- Sviluppo dell'applicazione in modalità scalabile e modulare, con riutilizzo di codice e quindi semplicità per future modifiche

4.2 Documentazione del software

I documenti che verranno prodotti durante lo sviluppo del progetto sono:

- Piano di Progetto
- Documento di analisi e specifica
- Piano di testing
- Documento di Progettazione

4.3 Funzionalità di supporto al progetto

Pianificazione della qualità

- Rispettare gli obiettivi e i requisiti (funzionali e non) definiti in corso d'opera
- Mettere in atto una severa politica di testing prima del rilascio della versione beta
- Utilizzare i feedback ricevuti dopo la pubblicazione della versione beta per il miglioramento dell'applicazione e rilasciare la versione ufficiale quando si ritiene che questa sia pronta

Pianificazione della gestione delle configurazioni

- Tracciamento di tutte le versioni dell'applicativo prodotto tramite il sistema di versioning GitHub, per permettere il ripristino in seguito ad eventuali modifiche

5. Pianificazione del lavoro, delle risorse umane e del budget

5.1 WBS (Word Breakdown Structure)

- A. Definizioni e idee di progetto: Si inizia a pensare ad un progetto valido e alle sue alternative (2 giorni)
- B. Definizione di gestione dei rischi: Si individuano tutti i possibili rischi e incombenze che possono portare a ritardi o perdita di lavoro (1 giorno)
- C. Definizione e preparazione degli ambienti di lavoro: Si pensa ai diversi ambienti di sviluppo da utilizzare, ai programmi da installare per il monitoraggio ed il testing dell'app e alle possibili location dove trovarsi per lavorare in gruppo (2 giorni)
- D. Decisione ruoli interni: Ci si consulta su come lavorare in gruppo, i compiti su cui ci si dovrà focalizzare, le responsabilità da prendersi e a quali componenti del gruppo affidare certe mansioni, sempre ricordandosi che questo gruppo lavorerà democraticamente cercando sempre di aiutarsi a vicenda per svolgere al meglio e senza ritardi le varie attività (1 giorno)
- E. Ricerca e studio del materiale informativo: Ci si informa dove poter trovare il materiale informativo, ad esempio le slide del professore o i manuali di utilizzo del lego mindstorm, scaricando e studiando il vario materiale, in modo da sapere con esattezza come poter progettare al meglio l'applicazione e il robot (2 giorni)
- F. Definizione struttura del Lego MindStorm: Si studia e ci si informa sulle componenti del Lego Mindstorm, su come utilizzarle e quali sono le possibili funzionalità da poter ideare in base alla struttura e robustezza del robot (4 giorni)
- G. Definizione funzionalità e interazioni con app: Si definiscono le funzionalità del robot e le interazioni utente-app e app-robot individuando i punti critici, le modalità per lo sviluppo e la suddivisione dei compiti (2 giorni)
- H. Stesura piano del progetto: Compilazione del documento seguendo le istruzioni date alle lezioni del prof. Agostino Cortesi (7 giorni)
- I. Analisi della fattibilità della costruzione: Si studia e si verifica che le funzionalità decise per il robot siano fattibili e coerenti, costruendo parti del robot per emulare e verificare che le componenti possano realmente funzionare. Se queste risultano non idonee, si procederà col definire funzionalità alternative (3 giorni)
- J. Costruzione del robot: Una volta decise e verificate le componenti e il loro possibile funzionamento si passa al montaggio completo del robot su cui si andrà a sviluppare l'app (21 giorni)

- K. Analisi di sviluppo interfaccia app-robot: Si studia come interfacciarsi al robot, decidendo le librerie da utilizzare e implementare (8 giorni)
- L. Analisi di sviluppo interfaccia utente-app: Si ricerca un modello U.I che sia chiaro e intuitivo per l'utente (4 giorni)
- M. Stesura documento di analisi e specifica: Si compila il documento seguendo le istruzioni date alle lezioni del prof. Agostino Cortesi (4 giorni)
- N. Sviluppo dell'applicazione Beta: Periodo di tempo dedicato all'implementazione della versione Beta (41 giorni)
- O. Stesura documento di testing: Compilazione del documento seguendo le istruzioni date alle lezioni del prof. Agostino Cortesi (6 giorni)
- P. Testing dell'app e del funzionamento del robot: Testing e verifica del corretto funzionamento dell'ecosistema app-robot (39 giorni)
- Q. Debugging: In contemporanea alla fase di testing si procede alla correzione dei bug rilevati sul codice (9 giorni)
- R. Stesura documento di progettazione: Compilazione del documento seguendo le istruzioni date alle lezioni del prof. Agostino Cortesi (5 giorni)
- S. Rilascio versione Beta (1 giorno)
- T. Analisi feedback Utente: Analisi di tutti i feedback ricevuti dopo la pubblicazione Beta (25 giorni)
- U. Sviluppo degli aggiornamenti: In seguito all'analisi dei feedback si svilupperanno gli aggiornamenti per il miglioramento complessivo del progetto (22 giorni)
- V. Testing e debugging sugli aggiornamenti: In contemporanea allo sviluppo degli aggiornamenti, si effettueranno attività di debug (22 giorni)
- W. Rilascio versione ufficiale (1 giorno)

5.2 Dipendenze

Diagramma di Gantt:
vedi allegato **GANTT.png**

Diagramma di Pert:
vedi allegato **PERT.png**

5.3 Risorse necessarie

Risorse Umane:

Sviluppatori, ossia i membri del gruppo.

In fase di Beta Testing sarà necessario radunare un campione di utenti per il test.

Risorse Hardware:

Computer personali di ogni membro del gruppo e alcuni dispositivi personali (solo quelli dotati di sistema operativo Android), set Lego Mindstorm EV3

Risorse Software:

IDE Android Studio e SDK Android installati su ogni computer utilizzato per lo sviluppo dell'applicazione.

Google Drive e GitHub per la creazione, condivisione e mantenimento delle versioni dei documenti e del codice sorgente.

Telegram e Discord come strumenti di comunicazione tra i membri del gruppo.

5.4 Allocazione del budget e delle risorse

Le risorse sono esclusivamente composte dai nostri dispositivi personali e dal Lego Mindstorm EV3. I software utilizzati per la realizzazione del progetto sono completamente gratuiti. Non sono dunque necessarie spese aggiuntive.

La seguente tabella è da non considerarsi in ordine cronologico.

Inoltre il calcolo dei giorni necessari per ogni attività non considera la sovrapposizione degli eventi.

ATTIVITÀ	GIORNI NECESSARI
Ideazione del progetto	14
Stesura documenti	22
Studio dell'interfacciamento tra applicazione e robot e modellazione dell'applicazione	15
Sviluppo applicazione Beta	41
Costruzione Robot	21
Fase di testing e raccolta feedback	74
Correzione bug e miglioramenti successivi al rilascio della versione Beta	44
Rilascio finale	1

Il periodo di tempo complessivo ipotizzato per l'ideazione e lo svolgimento del progetto, che va dal 09/10/18 al 31/01/19, corrisponde a 115 giorni totali compresi i fine settimana, corrispondenti a 32 settimane.

5.5 Pianificazione

- Piano di progetto 16/10/2018
- Documento di analisi e specifica 02/11/2018
- Piano di testing 15/11/2018
- Documento di Progettazione 10/12/2018
- Sviluppo e rilascio della versione Beta dell'applicazione 27/12/2018
- Rilascio della versione finale e documentazione 31/01/2019