



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

Retrospective Progetto WasteGone

Riferimento	C01_Retrospective
Versione	1.0
Data	17/12/2024
Destinatario	Docenti di Gestione dei Progetti Software 2024/25
Presentato da	C01 - D'Antuono Francesco Paolo, Fabiano Daniele
Approvato da	



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

Revision History

Data	Versione	Descrizione	Autori
08/12/2024	0.1	Prima Stesura	D.F.P. e F.D.
09/12/2024	0.2	Aggiunta dettagli nelle sezioni definite	D.F.P. e F.D.
17/12/2024	1.0	Revisione finale e consegna	D.F.P. e F.D.



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

Indice

Revision History	2
1. Main Informations	4
2. Project Description	4
3. Lessons Learned	6
3.1 Common Mistakes Checklist	6
3.2 Symptoms of Failure	6
3.2.1 Tecnologie utilizzate in maniera non ottimale	6
3.2.2 Testing delle componenti sottovalutato	6
3.2.3 Incapacità di rilevare errori negli artefatti	7
3.2.4 Processo di Code Review informale	7
3.2.5 Dispersione dei materiali e della comunicazione	7
3.2.6 Pianificazione dei meeting frettolosa	8
3.3 Recommendations for the Future	8
4. Post-mortem Review	9
5. Evaluation of Success/Failure	10



Retrospective del Progetto WasteGone

1. Main Informations

Titolo del progetto: WasteGone

Team ID: C01

Nome del team: GIGA

Data d'inizio del progetto: 18/10/2024

Consegna prototipo del progetto: 16/12/2024

Data fine del progetto: 28/03/2025

2. Project Description

L'obiettivo del progetto è fornire una piattaforma IT di supporto alle attività di gestione dei rifiuti, assicurando che tutti gli stakeholder coinvolti possano agire in maniera agevole ed efficiente. Per i cittadini, verrà realizzato un modulo AI che consenta di riconoscere il tipo di rifiuto, con le informazioni principali su dove e quando conferirlo.

Deve supportare le seguenti caratteristiche di base, che descrivono il sistema attuale:

- Creazione di un profilo per i cittadini;
- Visualizzazione dell'informativa sulle categorie dei rifiuti;
- Visualizzazione degli orari per il conferimento dei rifiuti;
- Notifiche per ricordare ai cittadini di conferire i rifiuti casalinghi (alla porta);
- Visualizzazione sulla mappa dei vari punti di ritiro per il conferimento di rifiuti speciali/ingombranti;
- Segnalazione per presenza di discariche abusive o errato conferimento dei rifiuti.



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

Sono previste anche le seguenti caratteristiche innovative:

- Prenotazione di uno slot per il ritiro a casa di oggetti/indumenti che se in buono stato saranno destinati alle donazioni, altrimenti saranno destinati allo smaltimento;
- Sezione eventi, per pubblicizzare eventi sul territorio che sensibilizzano sulle tematiche dell'ambiente e del riutilizzo;
- Sezione per effettuare baratto di prodotti che non sono più utilizzati;
- Sezione per effettuare pagamenti delle tasse sui rifiuti;
- Realizzazione modulo AI per riconoscimento del tipo di rifiuto.

La metodologia di sviluppo utilizzata è quella **Object-Oriented**, con Java come linguaggio di programmazione e Android Studio come ambiente di sviluppo, in modo tale da sviluppare un'app Native Android.

L'approccio utilizzato per lo sviluppo degli artefatti e, dunque, del prodotto è il **modello a cascata con feedback**, poiché è stato il modello più semplice e pratico da applicare per il contesto nel quale ci siamo trovati a lavorare. Infatti, i ragazzi hanno dovuto apprendere, prima di lavorare sul progetto, quindi non appena hanno appreso l'argomento, abbiamo potuto farli lavorare su quella parte specifica del prodotto.

La dimensione del progetto in LOC (lines of code) è:

```
[daniele@thinkbook14 app]$ cloc src/
 123 text files.
 122 unique files.
  41 files ignored.
```

github.com/AlDanial/cloc v 2.02 T=0.09 s (1320.3 files/s, 94848.4 lines/s)

Language	files	blank	comment	code
Java	88	1111	1387	3928
XML	34	222	105	2011
SUM:	122	1333	1492	5939



3. Lessons Learned

3.1 Common Mistakes Checklist

- **Tecnologie utilizzate in maniera non ottimale**
- **Testing delle componenti sottovalutato**
- **Incapacità di rilevare errori negli artefatti**
- **Processo di Code Review informale**
- **Dispersione dei materiali e della comunicazione**
- **Pianificazione dei meeting frettolosa**

3.2 Symptoms of Failure

3.2.1 Tecnologie utilizzate in maniera non ottimale

Questa problematica è stata individuata nel corso dello sviluppo del progetto, dove, in alcuni casi, si è avuta più difficoltà a realizzare alcune funzionalità.

La causa principale di questa problematica è la **scarsa conoscenza delle tecnologie**, la quale ha portato ad un approccio *learn-by-doing*, forzato dai tempi della consegna del prototipo.

3.2.2 Testing delle componenti sottovalutato

Questa problematica è stata individuata nel corso dell'implementazione del progetto, poiché spesso si è data più importanza alla realizzazione delle funzionalità, piuttosto che alla verifica del corretto funzionamento.

La causa principale di questa problematica è la **scarsa conoscenza delle tecniche di testing e dei tool** legati al medesimo. Infatti, il tempo allocato allo studio approfondito delle tecniche di testing, non è bilanciato per ciò che richiede la loro implementazione. Ciò porta ad avere casi di test banali e/o ridondanti.



3.2.3 Incapacità di rilevare errori negli artefatti

Questa problematica è stata individuata nel corso dello sviluppo degli artefatti, dove, in maniera frequente, c'è stato bisogno di ritoccarli e raffinarli.

La causa principale di questa problematica è la **mancanza di esperienza sull'ingegnerizzazione di un prodotto**. Gli artefatti che si devono produrre nel campo dell'ingegneria del software, richiedono un minimo livello di esperienza che, in questo contesto manca. Infatti, a volte, ci è voluto un po' più di tempo, per la produzione di alcuni artefatti.

3.2.4 Processo di Code Review informale

Questa problematica è stata individuata nel corso dell'implementazione del sistema software.

La causa principale di questa problematica è la **mancanza di ulteriore tempo per svolgere meeting appositi di ispezione del codice**. Le attività di review, dunque, sono state svolte, ma in maniera non controllata, ciò ha portato a scovare errori in maniera confusa e disorganizzata. Nonostante ciò, la code review ha agevolato la risoluzione di problematiche all'interno del codice.

3.2.5 Dispersione dei materiali e della comunicazione

Questa problematica è stata individuata nel corso dello sviluppo del progetto, poiché ci siamo trovati più volte in situazioni in cui era necessario condividere dei materiali ai Team Member e/o effettuare delle comunicazioni.

La causa principale di questa problematica è l'**utilizzo di più piattaforme per la condivisione dei materiali e per la comunicazione testuale/vocale, unita alla poca conoscenza di tool dedicati al project management**. Sarebbe stato conveniente individuare un'unica piattaforma attraverso la quale diffondere tutte le informazioni necessarie, in modo da avere un riferimento principale e una tracciabilità più semplice e immediata.



3.2.6 Pianificazione dei meeting frettolosa

Questa problematica è stata individuata nel corso dell'organizzazione dei meeting, poiché in alcuni casi le agende dei meeting sono state inviate troppo a ridosso della data dell'incontro.

La causa principale di questa problematica è **la difficoltà a trovare uno slot temporale compatibile con tutte le esigenze dei Team Member e dei Project Manager**. Sarebbe stato conveniente cercare di fissare fin dall'inizio, un calendario dei meeting, che seguisse passo passo la realizzazione delle milestone di progetto.

3.3 Recommendations for the Future

- **Attività di contorno schedate male:** Nelle fasi iniziali del progetto, si è temporeggiato in attività di brand identity, che non sono direttamente legate alla costruzione tecnica del sistema. Seppur queste attività hanno favorito la possibilità di un team-building più rapido, hanno diminuito il tempo a disposizione per le fasi successive del progetto.
- **Effettuare attività di training:** La scarsa conoscenza delle tecnologie potrebbe essere risolta con attività di training sulle tecnologie. Questo avrebbe aiutato a sviluppare più velocemente l'implementazione e il testing delle funzionalità.
- **Acquisire feedback esterni:** Sarebbe stato opportuno scambiare maggiori opinioni con gli altri Project Manager impegnati sui propri progetti, che avrebbero potuto fornirci feedback utili, per risolvere in maniera più rapida alcune problematiche.
- **Ricerca approfondita di tool di project management:** Non avendo conoscenza di molti tool relativi a questo ambito, in un futuro progetto si potrebbe investire del tempo ad inizio progetto per valutare e provare diversi strumenti di supporto alle attività di progetto.



Laurea Magistrale in Informatica - Università di Salerno

Corso di Gestione dei Progetti Software - Proff. F. Ferrucci, F. Palomba

4. Post-mortem Review

Per aver un punto di vista differente rispetto al nostro, è stato fornito ai Team Member un questionario, che consente di valutare l'operato di noi PM. I risultati del questionario sono disponibili al seguente [link](#).



5. Evaluation of Success/Failure

WasteGone	
Time	S
Product	S
Cost	S
Value	?
Use	?
Learning	S
Overall Stakeholder Satisfaction	?

Il progetto (fino al prototipo) ha rispettato i tempi e costi prestabiliti, incontrando tutte le esigenze di prodotto che erano state prefissate. Inoltre, anche i Team Member hanno appreso molti concetti nel corso dello sviluppo del progetto. Ciò non garantisce però il successo totale del progetto, poiché non è detto che esso verrà utilizzato e che il suo valore effettivo ci consenta di riuscire a rientrare dei costi totali del progetto. Tutti questi parametri permetteranno poi di stabilire se gli stakeholder sono soddisfatti o meno.