

# Post-retrospective

Riflessione di gruppo, team 8

## Sprechi identificati

Con le competenze sviluppate ora, il team è convinto di poter rifare il progetto in maniera molto più veloce, in quanto nella fase iniziale molte tecnologie e funzionalità (sia dell'applicativo in sé che delle metodologie di processo) erano nuove alla maggior parte dei membri:

- **Git** → capirne le funzionalità, i vari comandi e le diverse possibilità d'uso. Essendo una tecnologia nuova a tutti, inizialmente si è dedicato molto tempo a gestire i conflitti derivanti dalle diverse push di vari membri.
- **Django** → Capire come settare un ambiente condiviso a tutti i membri del team, evitando il caricamento di file che avrebbero potuto creare conflitti (tramite un .gitignore).  
Inizialmente è stato difficile trovare documentazione su certe funzionalità particolari che il team voleva sviluppare, il che ha portato alcuni rallentamenti.
- **Ajax** → difficoltà nella comprensione delle richieste Ajax, in particolare dei tipi di dato della richiesta e dei dati ritornati (JSON o meno); impiegato diverso tempo nel tentativo di comprendere le dinamiche Ajax e usarle al nostro servizio per gli scopi dell'applicativo.
- **Testing** → è stata necessaria una fase di documentazione approfondita prima di iniziare a scrivere veri e propri test in Django; questo ha rallentato l'implementazione dell'approccio TDD, in quanto inizialmente veniva scritto codice prima di creare test di verifica causa poca esperienza nell'ambito di testing.
- **Hosting cloud dell'applicativo** → difficoltà nell'inserire una applicazione Django su di un cloud gratuito, che offre una buona documentazione anche se un po' dispersiva, ma che ha imposto diversi limiti come la necessità di creare un ambiente virtuale su cui installare la versione di Django desiderata, un limitato spazio di memoria a disposizione sul server ed un limitato tempo di uso della CPU. Oltre tutto le impostazioni locali di Django (presenti sui nostri pc) non erano compatibili con quelle richieste dal server cloud. Essendo un servizio gratuito, non era possibile clonare il repository del nostro progetto direttamente da <https://dw.gnet.it/> perchè non era riconosciuto come host affidabile, per questo è stato necessario caricare

manualmente ad ogni nuovo rilascio di progetto, andando a cambiare i settaggi e ad inserire i comandi necessari. Tutto questo ha richiesto un buon quantitativo di tempo.

- **Metodologie agili** → essendo un argomento nuovo per tutto il team, all'inizio è stato dedicato molto tempo allo studio e alla piena comprensione delle tecniche da usare (Scrum, XP...). Anche i primi meeting e artefatti realizzati richiedevano molto tempo, e in alcuni casi erano anche troppo specifici rispetto a quanto richiesto.

L'organizzarsi in 4 sprint interni da una settimana ha però dato la possibilità al team di capire più in fretta come gestire al meglio i meeting e gli artefatti da produrre.

## Analisi dei limiti del sistema

I limiti del sistema, in questo momento, sono dettati da 2 aspetti: l'hosting dell'applicativo su cloud e l'applicativo in sè.

**Limiti dovuti all'hosting su cloud (<https://www.pythonanywhere.com>):**

- Limite massimo di **memoria**: 512 MB.
- Limite massimo di **tempo** di utilizzo della **CPU**: 100 secondi al giorno (dopodichè il processo viene eseguito più lentamente).
- L'applicazione viene eseguita in un **ambiente virtuale** (con i limiti prestazionali del caso).
- L'applicativo resta attivo sul server cloud per **soli tre mesi**, dopodichè esso verrà disabilitato. Per evitare ciò è necessario accedere al centro di controllo della web app almeno una volta nei tre mesi.

**Limiti dell'applicativo:**

- Massima **lunghezza** di un testo da analizzare di **5000 caratteri** (limite comunque modificabile ed aumentabile, settato a questo valore dopo dialogo con il cliente).
- Il database usato al momento è **SQLite**, un database appunto leggero che non permette un alto livello di concorrenza, non supportando un grande numero di utenti connessi contemporaneamente (ma è comunque molto stabile per un buon numero di utenti contemporanei).

La massima dimensione del database è di **140 TB**, un valore comunque abbastanza grande per l'applicativo sviluppato (facendo alcune prove, il team ha stimato circa 10KB per testo inserito, il che permette l'inserimento di un grande quantitativo di dati).

## Metodo effettivo di lavoro utilizzato


A inizio progetto, il team aveva scelto di usare **Scrum** con alcune tecniche di **XP**.

Tutti i membri concordano che il metodo di lavoro usato durante il progetto è stato rispettato, con l'utilizzo regolare di varie tecniche, tra le quali:

- Stand-up meeting giornalieri.
- Suddivisione in sprint di breve durata (una settimana circa per ogni sprint).
- Burndown charts per ogni sprint.
- User Stories e planning poker.
- Continuo dialogo e confronto con il cliente (soprattutto da parte del Product Owner, con partecipazione del team).
- Continui rilasci di MVP.
- Condivisione del lavoro e degli incontri con tutto il team, evitando eccessive specializzazioni.
- Vasto uso di pair programming.
- Refactoring frequente.
- Tentativo di applicazione del TDD nei test.
- Produzione degli artefatti tipici di Scrum, con sprint planning, retrospective, goals...

Il team si è anche cimentato in vari **quiz online** per testare il proprio livello di agilità, ricevendo sempre risultati molto vicini al livello di agilità ideale.

How you ask is everything [Create a typeform](#)



**How Agile Are You?**

Your score is 42

If your score is below 40, you should be looking to update your processes and team culture. Even if you score above 40, you should be looking to improve this score over time so it's worthwhile revisiting this 50 point test again in the future. Remember to share this to your team members to get a more accurate result of how your team's agility is performing.

Are you ready to have an effective Agile solution in your organisation? Finite IT can help you find the right people from your team from Product Owners, Testers, Managers and the Development Team.

[Quiz: How Agile Are You? - A 50 Point Test - Finite IT](#)

Il team è molto soddisfatto della scelta effettuata e non ritiene siano necessarie modifiche, in quanto ogni tecnica applicata è risultata molto efficace (ad esempio, il suddividere il progetto in quattro sprint da una settimana ciascuno invece di due sprint da due settimane ha permesso di organizzare il lavoro in modo molto più chiaro e ha agevolato l'acquisizione di esperienza sia di programmazione che della metodologia agile).