s1retrospective

Introduzione

Arrivando alla fine del primo sprint, il team si è incontrato in data 9/11/23 (nello stesso giorno della sprint review) per discutere dell’andamento comportamentale dei componenti, rispetto alla realizzazione di quanto stabilito.

Il principale problema riscontrato, riconosciuto da tutti, è stata la poca organizzazione del team, dovuta anche al fatto che questo è stato il primo effettivo sprint in cui l’obiettivo era una release pratica del prodotto, che quindi implicava la necessità di scrivere codice.

Questo ha fatto sì che impedimenti minori come piccole lacune di alcuni membri con le architetture e i linguaggi utilizzati (seppur ora colmate a dovere), impegni personali di ogni componente e poca comunicazione sono diventati rilevanti, portando a piccole frizioni e attriti nell’andamento del lavoro.

Tuttavia, nonostante quanto appena detto, il team è riuscito con successo ad implementare più di quanto previsto, mantenendo anche una certa eleganza nell’interfaccia grafica, seppur minimale, e fornendo una versione della webapp che implementa una prima bozza della variante Kriegspiel, perfettamente giocabile, nonostante il gruppo abbia intenzione di rivedere l’arbitro.

Punti di discussione

- necessaria una più attiva partecipazione alle riunioni (daily scrums) e nelle sessioni di preparazione del lavoro
- serve più collaborazione sul codice, anche non scrivendolo direttamente ma comunque essendo presenti durante sua stesura
- servono aggiornamenti del VCS più costanti, altrimenti è chiaro che si crea una situazione di "ostaggio" nei quali alcuni developers sono impossibilitati a modificare un codice non pushato
- più conformità nella gestione delle US su Taiga, non arrivando ad un punto in cui alla fine del lavoro vengono spostate tutte direttamente su ready

- migliore suddivisione del lavoro fra i componenti, da effetturare appena decise le US per lo sprint, nonché decidere tempestivamente la priorità e le stime delle US

Domande di rito

Cosa è andato bene / Cosa ti è piaciuto?

- Chiarini (Product Owner) → risultato finale
- Patella (Scrum Master) → risultato finale, onestà del team nella retrospettiva
- Busnelli (Developer) → implementate funzionalità minori non richieste ma necessarie
- Folli (Developer) → contento del lavoro di SM e PO e del risultato finale
- Testa (Developer) → mole di task effettivamente concluse e risultato finale
- Urbano (Developer) → complessivamente nulla di importante, solo il fatto di aver concluso correttamente lo sprint

Cosa è andato male / Cosa non ti è piaciuto?

- Chiarini (Product Owner) → organizzazione povera, i developer non hanno prestato la dovuta attenzione al backlog
- Patella (Scrum Master) → organizzazione povera, pochi aggiornamenti del VCS, poca iniziativa dei developer, insoddisfatto del proprio lavoro
- Busnelli (Developer) → organizzazione povera, backlog non sempre consistente
- Folli (Developer) → organizzazione povera, insoddisfatto del proprio lavoro
- Testa (Developer) → organizzazione povera, daily scrum quasi assente e sprint planning non collaborativo, insoddisfatto del proprio lavoro
- Urbano (Developer) → organizzazione povera, poco aiuto dal team

Cosa si deve fare di diverso?

- Chiarini (Product Owner) → organizzarsi meglio, prestare più attenzione al backlog
- Patella (Scrum Master) → organizzarsi meglio, aggiornare di più il VCS e la board

- Busnelli (Developer) → gestire meglio la priorità delle task da svolgere
- Folli (Developer) → maggior comunicazione e organizzazione
- Testa (Developer) → gestire meglio le task e i cambiamenti delle US sul backlog
- Urbano (Developer) → maggiore partecipazione del team, migliore divisione delle task

---

Buoni propositi

Iniziare a:

- scrivere la documentazione appropriata del codice
- fare commits ogni volta che una task è stata completata

Smettere di:

- scrivere codice senza consultare e aggiornare il backlog dello sprint
- fare commits troppo pesanti

Continuare e migliorare:

- arricchire il file di Sprint Goal, Definition of Done e Definition of Ready

Fare di più:

- distribuire meglio il carico di lavoro, in altre parole organizzare meglio le task
- daily scrums
- pair programming

Fare di meno:

- ritardare l'organizzazione delle US