

SPRINT 2 RETROSPECTIVE

Per la seconda retrospettiva è usato il metodo con alcune delle carte Essence dei sottoinsiemi: Alpha, Scrum essentials e Agile essentials

	Great	Ok	Not so good	Improvement actions
High				1) Le prossime conversazioni sui requisiti con gli stakeholder dovranno essere dettagliate e precise 2) Divisione più specifica delle US
Medium				1) Creazione di Unit test != da Integration test 2) Meno codice hard codato
Low				

Cosa ha funzionato:

- Il team risulta coeso e flessibile per affrontare le difficoltà riscontrate grazie alla disponibilità di tutti i membri e l'utilizzo predominante del pair programming
- Tutta la documentazione è stata modificata in base ai feedback ricevuti, secondo il metodo agile, in particolare focalizzandosi sul Backlog, spezzando le US in altre più dettagliate (presente nel sprint_1 retrospective)
- La modalità di programmazione in pair programming
- Ogni membro del gruppo comunica tempestivamente gli upgrade effettuati
- Riunioni quotidiane per scambiarsi opinioni e feedback, utilizzando appositi tools, soprattutto dopo incontri con stakeholders
- Sono stati studiati e implementati programmi per il controllo dello sviluppo del progetto come: Mattermost, Jenkins e SonarQube (presente nel sprint_1 retrospective)

- Utilizzo più professionale e distribuito di Git (presente nel sprint_1 retrospective)

Cosa poteva essere migliorato:

- Meno codice hard codato e scrittura di codice più flessibile ai cambiamenti, in linea con lo sviluppo agile
- Ulteriore partizionamento delle User Stories in task, per rendere più chiaro e immediato l'avanzamento del progetto.
- Sono stati realizzati test dove c'era un ritorno importante per verificare la corretta azione delle funzioni osservate, purtroppo si sono rilevati Integration test, quando erano stati richiesti Unit test (presente nel sprint_1 retrospective)
- Maggior comprensione dei requisiti richiesti per evitare di compiere lavoro inutile

Cosa si può fare per migliorare nel prossimo sprint:

- Migliorare la comunicazione tra i colleghi sfruttando Mattermost a partire da adesso
- Utilizzo delle informazioni e statistiche date da Jenkins e SonarQube per migliorare la qualità del codice una volta che si è arrivati a soddisfare gran parte dei requisiti
- Revisione periodica della documentazione a partire dai prossimi feedback
- Richiesta di ulteriori riunioni e chiarimenti se necessario a partire da questo sprint