Rapporto del TEAM 14

Andreea Scrob,

Yonas Ernesto Catalini, Luca Tagliavini,
Gaia Clerici, Gianmaria Rovelli, Francesco Apollonio

December 2022

Indice

1 Descrizione del prodotto

Bluebird non è solo un sito web in cui si possono visualizzare dei tweet, bensì un servizio che offre molto di più. Parliamo ad esempio di un'analisi del sentimento del tweet, una classifica di coloro che provano a indovinare su Twitter la parola del gioco "la ghigliottina", parte della trasmissione "*l'Eredità*", o ancora di una pagina per chi vuole giocare a scacchi con più persone contemporaneamente e molto altro ancora.

Nella prima pagina possiamo cercare una qualsiasi parola oppure un nome utente: il sito ci darà in risposta una lista di tweet, provvedendo per ogni messaggio l'esito di un'analisi dei sentimenti (*Sentiment Analysis*) sul contenuto dello stesso. Quest'analisi è svolta da una IA (*Intelligenza Artificiale*) e i dati ottenuti vengono inoltre raccolti nel grafico a torta che possiamo notare sulla sinistra della pagina. Accanto ad esso si trova un ulteriore grafico a barre che indica quanti tweet sono stati raccolti in relazione al tempo. Subito sotto abbiamo invece una Word-Cloud che raccoglie le parole più utilizzate. La dimensione di queste varia in base alla loro presenza nei risultati ottenuti dal social network. Abbiamo infine una mappa in cui vengono contrassegnati i tweet provvisti di geolocalizzazione.

Inoltre, troviamo le funzioni dedicate alle trasmissioni televisive. La pagina dell'Eredità permette di visualizzare tutti i tweet con "#ghigliottina " e "#eredita", un podio di coloro hanno indovinato la parola del giorno nel minor tempo possibile, un grafico che indica quante persone hanno indovinato la parola e una nuvola di parole che corrispondono alle risposte proposte dagli utenti.
Diversa è invece la pagina di Fantacitorio: un servizio che fa riferimento al programma televisivo "*Propaganda*" in onda su LA7. In questa pagina troviamo la raccolta di immagini delle squadre proposte dagli spettatori e una classifica con i punti cumulatividi ciascun politico. Qui è possibile inoltre cercare un determinato utente e visualizzare, se esiste, l'immagine della sua squadra.

Infine è presente la sezione Scacchi: una pagina dedicata agli amanti più sfegatati di uno dei più famosi giochi di strategia pronti a cercare un'esperienza diversa dal solito. La particolarità qui consiste nel giocare contro più di una persona; chiunque potrà, infatti, twittare la propria mossa rispondendo allo stato della partita pubblicato dal servizio. Successivamente la mossa più quotata, se ammissibile, verrà giocata. Tutto ciò viene registrato da un grafico in modo da avere una visione più immediata delle mosse scelte. Il giocatore che ha avviato la partita su Bluebird potrà decidere la durata dei turni e visualizzare le mosse twittate dagli avversari direttamente dalla nostra piattaforma.

1.1 Scope e product backlog

Scope È l'insieme dei processi necessari a garantire che l'ambito di un progetto sia definito e mappato in modo preciso. Comprenderebbe un analisi approfon-

dita dei requisiti per poi preparare una progettazione del lavoro dettagliata.
Essendo un progetto universitario che simula la realtà, si partiva già da una descrizione del prodotto.
Per assecondare le richieste dei clienti il progetto era già diviso in epiche: le prime e più generiche linee guida da seguire per lo sviluppo corretto del prodotto finale. Le epiche, una volta iniziato lo sviluppo del progetto, vengono suddivise in oggetti più piccoli e gestibili (le User Stories). In questo caso le epiche finali sono risultate differenti da quelle iniziali; questo è dovuto al variare continuo delle richieste da parte del cliente. Seguono le epiche iniziali fornite dal cliente.

Il prodotto deve poter supportare le seguenti attività:

1. *Come spettatore de #leredita (RAI1):*
   voglio raccogliere i tweet di chi prova a indovinare la ghigliottina, per visualizzare (in ordine temporale, o su una mappa) tutti coloro che indovinano.

2. *Come spettatore di #reazioneacatena (RAI1):*
   voglio raccogliere i tweet di chi prova a indovinare l'ultima parola, per visualizzare tutti coloro che indovinano.

3. *Come giocatore di scacchi:*
   voglio sfidare gruppi di persone in rete, per giocare partite le cui mosse verranno scelte a maggioranza.

4. *Come giocatore di scacchi:*
   voglio raccogliere i tweet che rispondono alla mia mossa, per scegliere e visualizzare la mossa scelta dalla maggioranza.

Utilizzare il metodo agile significa poter modificare in stato di processo il prodotto richiesto, grazie a un rilascio frequente e un confronto diretto con il cliente.Per questo motivo le epiche sono variate con il tempo.

- Pagina di ricerca:
  Come utente vorrei una pagina di ricerca con strumenti annessi.

- Fantacitorio:
  Come visitatore, voglio poter accedere ad una sezione nel sito web fatta appositamente per mostrare e cercare informazioni riguardanti il Fantacitorio.

Il grafico in Fig.1 notiamo l'andamento del progetto in tutto periodo di sviluppo che va dal 18 Ottobre al 18 Dicembre. Possiamo notare come la distribuzione del lavoro sia quasi lineare durante ciascuno dei quattro sprint.

1.2 Diagrammi dei casi d'uso

La figura 2 rappresenta un diagramma che esprime un comportamento, offerto o desiderato, sulla base dei suoi risultati osservabili. Questo diagramma non è stato creato così dal primo incontro, bensì è l'insieme di tutti i diagrammi

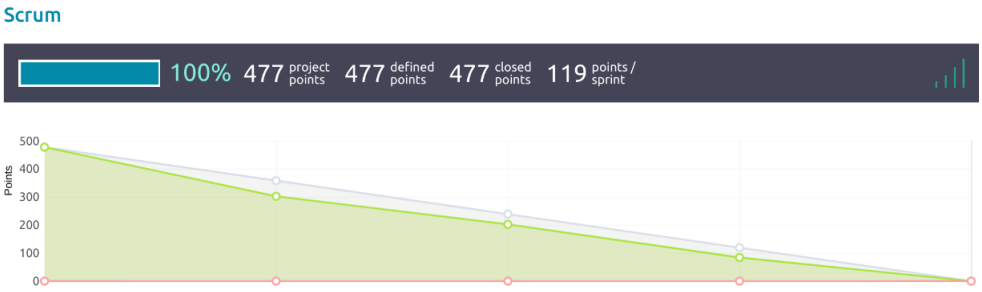

Figura 1: Grafico presente Backlog del progetto

fatti a ogni analisi dei requisiti. Essi servivano principalmente al Team per visualizzare il lavoro da svolgere per ogni epica, e di conseguenza scegliere il da farsi per lo sprint corrente. Nella primo sprint l'obbiettivo era quello di prendere confidenza con gli strumenti che si sarebbero utilizzati in futuro come l'api di Twitter, il linguaggio di programmazione scelto, ecc.... Per questo motivo è stato opportuno fare una prima stesura di quello che ora è il ramo dedicato alla **Ricerca**. Similmente è successo per tutti gli altri sprint. Le immagini venivano poi caricate nella wiki di taiga del team, affinché fossero a disposizione di tutti.

2 Sprint 0

Per iniziare a prendere confidenza con la metodologia Agile, ai team è stato chiesto di giocare una partita a Scrumble. Questo gioco è la rappresentazione verosimile di quello che nella realtà si affronta durante lo sviluppo di un progetto. Permette, infatti, di prendere confidenza col mondo del lavoro e con gli strumenti che lo caratterizzano: user stories, task, debito tecnico e persino con il concetto stesso di Sprint. Il tutto si basa su una giusta divisione del lavoro e una corretta assegnazione dei ruoli. Quest'ultimo è la nozione pilastro su cui è fondato il gioco; un'assegnazione sbagliata potrebbe, infatti, portare a una "perdita della partita" che nel mondo del lavoro si traduce in una vera e propria sconfitta lavorativa. Questa potrebbe essere dovuta ai più svariati motivi come la mal gestione del carico di lavoro, un utilizzo improprio del tempo a disposizione o una scorretta comunicazione all'interno del team.

Il team in questione ha svolto due partite di Scrumble: la prima è stata un fallimento totale, lo scrum master Luca non aveva ben compreso le dinamiche del gioco il che si è rispecchiato sullo sviluppo della partita risultata problematica. Yonas aveva svolto il ruolo del PO, inizialmente titubante ha provato a scegliere le User Stories, nei primi sprint la cosa gli è risultata un po' difficile, me già dal terzo in poi il gioco è diventato più chiaro e di conseguenza le deci-

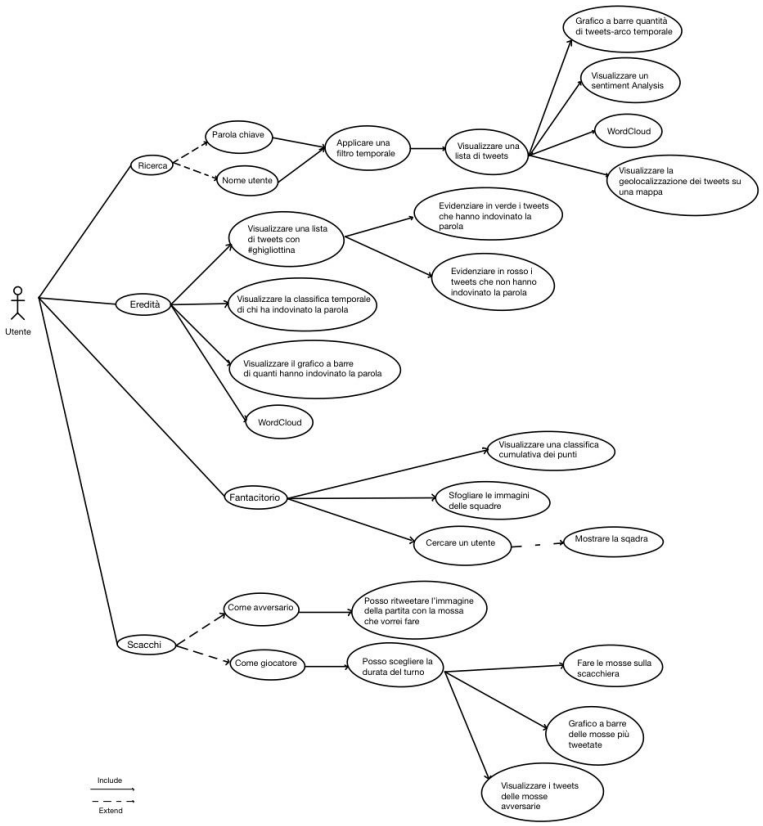

Figura 2: Diagramma dei casi d'uso del progetto.

sioni prese risultavano sempre più adatte al team. Il gioco non si è totalmente concluso per mancanza di tempo. Si era tuttavia giunti a una situazione insostenibile causata del debito tecnico nato dalle pessime decisioni prese dal team nei primi turni.

Nei giorni successivi al team si è aggiunta una nuova persona sfatando l'insegnamento appreso a lezione dalla legge di Brooks, che recita:

> *Aggiungere forza lavoro ad un progetto software in ritardo, lo farà ritardare ancora di più.*

Questa aggiunta non è stata, infatti, un particolare problema e non ha causato nessun tipo di ritardo. Probabilmente questo perché la legge prende in considerazione progetti che sono già in un pessimo stato di avanzamento. Ciò ha tuttavia obbligato i ragazzi a giocare una nuova partita. Vista la precedente, è stato saggiamente deciso di cambiare i ruoli, quanto più quello dello Scrum Master dato che il PO, alla fine della partita precedente era riuscito ad entrare nell'ottica di un buon analizzatore delle capacità del team e delle proprie responsabilità. Anche questa volta non si è concluso il gioco ma si è giocato quanto basta per capire quella sarebbe stata l'organizzazione definitiva. Concluso lo sprint le vesti sono state:

- Andreea Scrob come Scrum Master (**SM**);
- Yonas Ernesto Catalini come Product Owner (**PO**);
- Gaia Clerici, Gianmaria Rovelli, Luca Tagliavini e Francesco Apollonio come Developer Team.

Ovviamente anche Andreea e Yonas quando non sono in funzione di SM e PO fanno parte del Developer Team.

2.1 Valutazione

Alla fine dello sprint 0 è stato richiesto a ogni componente del team di rispondere ad alcune domande relative al gioco. Dalla valutazione ne risulta che l'ultima partita giocata, quella che poi è stata realmente votata dai componenti, è andata nel complesso bene. I ragazzi risultano aver capito le regole, anche se rimasti un po' perplessi a seguito della prima partita. A ogni sprint si sono concluse circa 5 user stories, di diversa difficoltà, che sono state scelte dal PO. Il team ha collaborato e spesso i ragazzi si confrontavano tra di loro. Sono state rispettate le pratiche della sprint review e della retrospettiva. La sprint review che ricordiamo essere una revisione finale dello sprint volta a verificare che l'obiettivo prefissato sia stato raggiunto ed eventualmente con quali risultati. A questo processo ha partecipato tutto lo Scrum Team e solitamente partecipa anche il committente del prodotto a cui viene mostrato il lavoro svolto fino a quel momento. La sprint retrospettiva è un'ulteriore analisi effettuata con la partecipazione di tutto lo Scrum Team utile a valutare cosa continuare a fare,

GOAL	QUESTIONS	EVALUATION	Andreea Scrob	Yonas Catalini	Gaia Clerici	Gianmaria Rovelli	Luca Tagliavini	Francesco Apollonio
Learn	Q1	1 = no idea of the Scrum roles 5 = perfect knowledge of the roles and their jobs	5	4	3.7	4.3	4.7	4
	Q2	1 = couldn't repeat the game 5 = could play the game as a Scrum Master by himself	4	3	4	3.8	4.5	3.5
	Q3	1 = totally lost 5 = leads the game driving the other players	4	4	3	5	5	4
Practice	Q4	1 = feels the game is unrepeatable 5 = feels the game could be played in any situation	4	5	4.5	4.7	4.5	5
	Q5	1 = 0 to 3 stories 2 = 4 to 6 3 = 7 to 9 4 = 10 to 12 5 = 13 to 15	4	4	4	4	4	4
	Q6 ONLY DEV TEAM	1 = abnormal difference from the other players 5 = coherent and uniform with the group most of the time			4	5	4.8	4.5
Cooperation	Q7	1 = never speaks with the other players 5 = talks friendly to anyone in every situation	5	5	5	5	5	5
	Q8	1 = never puts effort in doing something 5 = every time is willing to understand what is going on	3	4	5	5	4.5	4
	Q9	1 = never asks for an opinion 5 = wants to discuss about every topic	5	4.7	4	4.8	5	4.3
Motivation	Q10	1 = not involved by the game 5 = always makes sure everyone is on point	5	5	4	5	5	5
	Q11 ONLY FOR PO	1 = poor/absent advices 5 = wise and helpful suggestions when is required		4				
	Q12	1 = doesn't express opinions during retrospective 5 = feels the retrospective fundamental to express opinions	3	4	3	4	3	4
Problem Solving	Q13	On the game board, if the debt pawn is on the lowest stage, the evaluation is 5, for every higher stage it decreases by 1	3	3	3	3	3	3
	Q14 ONLY DEV TEAM	Calculate the average of tasks left for each sprint: 1 = 21+ 2 = 16-20 3 = 11-15 4 = 6-10 5 = 0-5			5	5	5	5
	Q15 ONLY FOR PO	Same evaluation as Q14 for the PO		4.5				

Figura 3: Valutazione della partita a Scrumble fatta.

cosa no e cosa migliorare nello sprint successivo per ottenere performance ancora più efficienti. Queste pratiche sono state utili per l'organizzazione del lavoro sia nel gioco che, successivamente, nel progetto vero e proprio; ciò ha permesso al PO di comprendere come scegliere delle User Stories e dividerle in task. Infatti, nel gioco era riuscito a svolgere i suoi compiti in maniera tale da non causare accumulo di lavoro arretrato.

3 Sprint 1

Data inizio: 18 Ottobre 2022
Data fine: 2 Novembre 2022

Il team si è riunito per nei primi giorni dello sprint per decidere come strutturare il lavoro, visualizzare le specifiche del progetto e fare un analisi dei requisiti. Si sono esaminati gli strumenti proposti e si è deciso di utilizzare:

- Gitlab: come piattaforma web per mantenere il codice;
- Taiga: per la gestione del lavoro;
- Sonarqube: come piattaforma per l'ispezione continua della qualità del codice;
- Mattermost: come mezzo di comunicazione.

Oltre alla divisione in Scrum Muster (Scrob), Produc Owner (Catalini) e Developers, i ragazzi si sono divisi nella gestione del Back-End (Tagliavini, Rovelli, Apollonio) e Front-End (Scrob, Clerici, Catalini). Per far fronte al gioco lo SM ha deciso che i DailyScrum si sarebbero svolti il Martedì e il Giovedì, in modo da riuscire a tenere sotto controllo l'andamento delle task.

3.1 Sprint Goal

L'obiettivo era quello di fare una stesura iniziale delle User stories e strutturare una pagina iniziale in cui visualizzare i tweet ottenuti da Twitter.

3.2 Sprint backlog

Le User Stories scelte per questo sprint non erano molte; come accennato, infatti, l'obbiettivo era quello comprendere lo svolgimento generale del lavoro e come utilizzare al meglio gli strumenti. Le US proposte dal PO sono state:

1. Visualizzare tweet;
2. Filtrare per parola;
3. Filtrare per persona;
4. Andare al profilo dell'autore;
5. Andare al tweet originale;
6. Mostrare la mappa;
7. Salvare e mostrare richieste precedenti;
8. Filtrare per tempo.

Ognuna di esse veniva divisa in task, le quali, a loro volta, venivano assegnate a uno o più componenti del team. Questi ultimi avevano l'incarico di valutare le US in base alla difficoltà e alla quantità di tempo stimato per portarle a termine. Dunque, veniva decisa una 'data di scadenza' delle User Stories e, in prossimità di questa, lo Scrum Master aveva l'incarico di controllare il lavoro svolto; in questo modo poteva eventualmente aiutare chi fosse stato in difficoltà oppure spronare chi non stesse lavorando. Il tutto si può visualizzare nella Fig. 4.

3.3 Definition of Done

Essendo il primo sprint di lavoro effettivo, il team ha dovuto concordare insieme al PO sulla loro **Definition of Done**, ovvero il criterio con cui il PO può ritenere un determinato lavoro completo. Si è quindi deciso che quando il codice rispecchiasse ciò che era scritto nella US ed era stato sufficientemente testato, allora si poteva considerare completata la User Story.

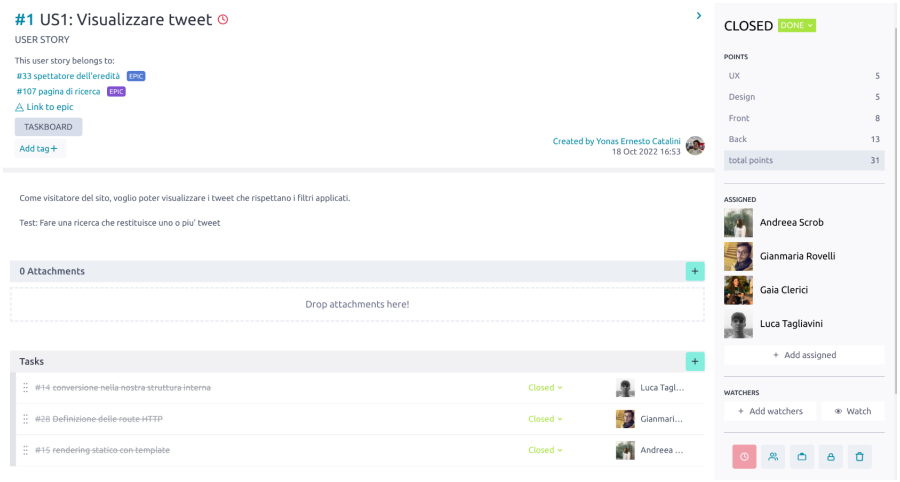

Figura 4: Organizzazione di una User Story.

3.4 Test

In questo sprint non erano richiesti particolari test ma nonostante ciò, il team, indotto dalla Definition of Done, li ha eseguiti. Il gruppo ha deciso di implementare una cache per poter avere un ampio spettro di dati per la successiva implementazione dei grafici. Di seguito la lista di descrizioni delle User Stories e dei test annessi:

1. Come visitatore del sito, voglio poter visualizzare i tweet che rispettano i filtri applicati.
   Test: Fare una ricerca che restituisce uno o più tweet.

2. Come visitatore, voglio poter filtrare i tweet mostrati secondo una parola chiave.
   Test: cercare i tweet contenenti "#IngSw2022", e tutti i tweet contenenti "933q48phtagui;fwa;oijref09823".

3. Come visitatore, voglio poter visualizzare dei tweet di uno specifico account.
   Test: Richiedere tweet di @MatteoSalvini e @AccountInesistente1337.

4. Come visitatore, voglio poter visualizzare dei tweet geolocalizzati su una mappa.
   Test: Aprire sito, visualizzare la mappa generica, e generata in base ai tweet di uno specifico account.

5. Come utente, voglio che mi vengano mostrati non solo risultati della richiesta corrente, ma anche risultati rilevanti di richieste passate salvati dal servizio.

Test: Dopo aver popolato la cache (i.e. con una ricerca), ottengo risultati senza ricontattare Twitter.

6. Come visitatore, voglio poter visualizzare tweet pubblicati in una specifica fascia di tempo.
   Test: Visualizzare tweet di Ottobre 2022, e di Agosto 1945.

3.4.1 Sonarqube

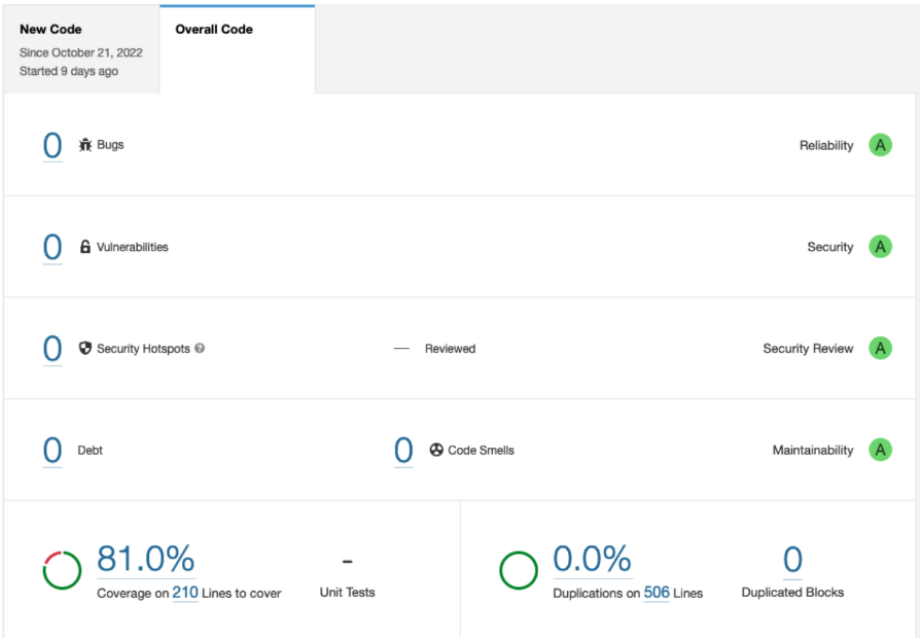

Figura 5: Dati di Sonarqube ricavati a fine dello Sprint 1.

Nel primo sprint si è implementata la maggior parte di quello che si era prefissato mantenendo una buona coverage del codice. È possibile che questo testing aggressivo fin dall'inizio abbia rallentato il team ma, col senno di poi, ha in realtà costretto a rivalutare la struttura del codice Back-End e scegliere una serie di interfacce abbastanza generali fin da subito. Anche alla fine del progetto questa implementazione non è stata alterata, mostrando dunque che il tempo inizialmente dedicato al decidere una buona forma del codice ha ripagato a lungo andare.

Già da questo primo sprint si nota un pattern comune in tutti i nostri report di coverage: non si riuscirà mai a coprire al 100% tutte le linee di codice scritte a causa di come funziona il testing in Go. Per ottenere una coverage perfetta, infatti, si necessita di entrare in tutti i rami d'errore possibili. Alcuni di questi sono casi rari che non solo è pressoché impossibile incontrare nella pratica, ma

è anche difficile riprodurre "in laboratorio" per fare del testing (i.e. il fallire a validare degli URL hard-coded). Questa realtà implica che la percentuale di copertura non raggiungerà mai il massimo, ma non va interpretata come una mancanza di test.

3.5 Burndown

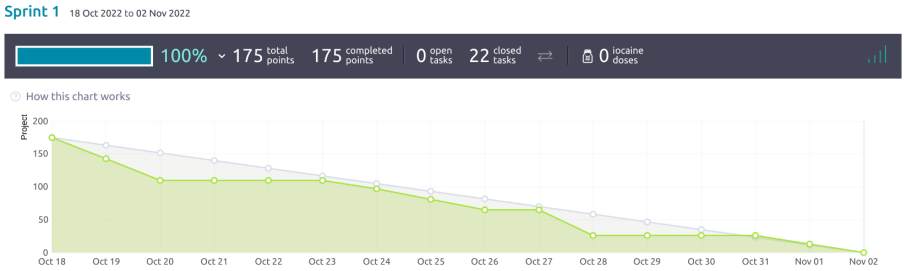

Figura 6: Grafico Burndown dello Sprint 1.

Come si può notare dal grafico in figura 6 il team è stato molto produttivo in corrispondenza dei Daily Scrum (precedentemente stabiliti dallo Scrum Muster). Questo accadeva in quanto essendo tutto il gruppo riunito, si riusciva a discutere meglio per quanto riguardava le decisioni da prendere, ma soprattutto si riusciva a dare una mano a chi fosse bloccato su determinati problemi. In quei giorni, infatti, si svolgeva principalmente *pair programming* che velocizzava notevolmente la stesura del codice e ne migliorava la qualità. Di seguito un piccolo resoconto degli incontri effettuati:

- 20/10/2022:
  Il PO ha proposto le US dello sprint, le quali sono state divise in task e votate. Ogni membro, sotto consiglio di Yonas, ha scelto le task da fare ed ha iniziato a lavorarci.

- 25/10/2022: Durante questo meeting si è fatto un punto della situazione per vedere il lavoro svolto da ogni developer. Nel complesso si aveva una pagina web in cui si potevano visualizzare i tweet. Erano state iniziate anche le User Stories riguardanti la ricerca per username e per keyword.

- 27/10/2022: Il gruppo si è riunito per risolvere degli intoppi dovuti alla visualizzazione della mappa. Mentre le ragazze del front-end lavoravano alla mappa, Tagliavini e Rovelli si occupavano della cache. La giornata si è conclusa con un pair programming tra Apollonio e Tagliavini per i test della cache.

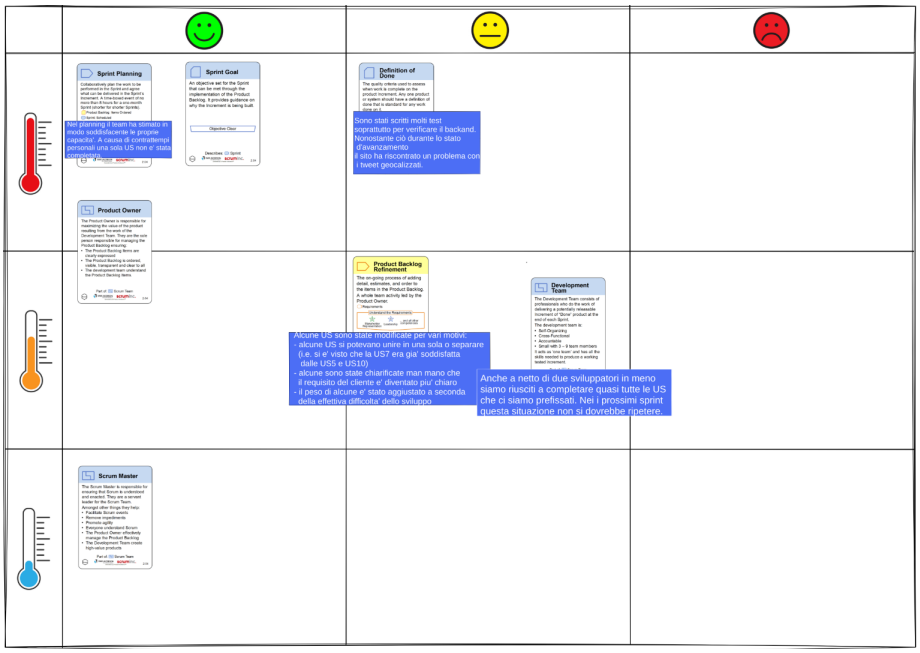

Figura 7: Retrospettiva dello sprint 1.

3.6 Retrospettiva

Nella figura 10 troviamo i seguenti commenti: *"Nel planning il team ha stimato in modo soddisfacente le proprie capacità. A causa di contrattempi personali una sola US non è stata completata."* e sulla **Definition of Done** *"Sono stati scritti molti test soprattutto per verificare il Back-End. Mancano alcuni test per i Front-End.* Questo è stato modificato in seguito, perché durante lo stato d'avanzamento il Professore ha fatto cercare "juventus" e il sito ha interrotto la sua esecuzione a causa dell'ultimo aggiornamento in cui era stata inserita la mappa. La causa era dovuta a Twitter che non aveva aggiornato la documentazione su come gestire i tweet geolocalizzati. Il commento definitivo riportato sulla card è: *"Sono stati scritti molti test soprattutto per verificare il back-end. Nonostante ciò durante lo stato d'avanzamento il sito ha riscontrato un problema con i tweet geocalizzati.*
Per il **Development Team** si è sentita la necessità di spronare i componenti del team a essere più possibile presenti e partecipativi, per evitare che chi fosse impossibilitato a essere presente fisicamente nelle attività non si distaccasse e si demotivasse producendo meno per il prodotto. L'ultimo commento è principalmente tecnico, piccoli accorgimenti riguardo la stesura delle User Stories che sono emersi solo lavorandoci, come: accorpare due User Stories ne evitava una terza, alcune User Sotries erano poco chiare, altre invece erano sopravvalutate.

3.7 Considerazioni finali

Al primo riscontro con lo stakeholder il gruppo ha presentato una pagina web in cui si poteva fare una ricerca per nome utente e per parola chiave, in base a cui si ottenevano i tweet relativi. Il sito presentava funzionalità non richieste, mancavano invece strumenti come il filtro temporale che erano più rilevanti. Questo è stato un errore importante per il team, perché gli ha fatto capire che c'è una priorità tra le User Stories e che lo scopo del progetto era diverso da altri svolti precedentemente, dove l'obiettivo principale era quello di rendere il prodotto il più piacevole possibile all'utilizzo, anche implementando funzionalità non specificate.

4 Sprint 2

Data inizio: 3 Novembre 2022
Data fine: 16 Novembre 2022

Considerato l'errore commesso nello sprint precedente, i ragazzi hanno deciso di procedere diversamente. Nel primo Daily Scrum è stata svolta un'altra analisi dei requisiti, valutando cosa si sarebbe potuto concludere nelle successive settimane, giacché il PO era via per motivi personali e che quindi avrebbe dovuto lavorare da remoto. Si è così iniziato a lavorare alle funzionalità essenziali e obbligatorie come il grafico a torta e la Word-Cloud. Dopo una casuale conversazione che lo Scrum Muster ha avuto con il professore, ha deciso insieme al team di assemblare un mock-up da proporre al cliente in modo da evitare incomprensioni.

4.1 Sprint Goal

L'intento di questo Sprint era quello di completare le funzionalità richieste dalle specifiche del progetto, ovvero: Sentiment Analysis, grafico a barre indicante il numero di tweet trovati in un giorno e una Word-Cloud.
Inoltre si era prefissato anche di iniziare a implementare la pagina dell'eredità.

4.2 Sprint backlog

Dato lo sprint goal le user sotries sono state:

1. Word Cloud;
2. Sentiment Analysis;
3. Grafico a torta dei sentimenti;
4. Diagramma a barre;
5. Sito mobile;
6. Gestione errori.

Apparentemente sembrerebbe che in questo sprint siano state scelte meno US, in realtà la complessità di queste risultava maggiore di quelle delle settimane precedenti. Esse sono anche state divise seguendo il metodo *Moscow* (è l'acronimo di Must, Should, Could e Won't), utilizzata per avere una priorità tra di loro. Le prime quattro fanno parte della sezione **Must**, la quinta della **Should** e l'ultima di **Could**.

4.3 Definition of Done

La Definition of Done è rimasta invariata rispetto allo sprint precedente, di conseguenza i developers potevano considerare le loro task complete solamente dopo essersi accertati che il codice scritto funzionasse e che fosse sufficientemente testato.

4.4 Test

In questo paragrafo sono state riportate le spiegazioni delle User Stories scelte per lo sprint con i test:

1. Come visitatore, voglio poter visualizzare una "nuvola di parole" generata in base ai tweet visualizzati.
   Test: Generare una Term Cloud dei tweet di @elonmusk

2. Come visitatore, voglio poter visualizzare il "sentimento" di un gruppo di tweet.
   Test: Controllare se chi guarda #familyfeud è soddisfatto.

3. Come visitatore, voglio poter visualizzare un grafico a torta che mostra la percentuale di tweet positivi e negativi a partire da una sentiment analysis.
   Test: Visualizzare il sentimento degli utenti riguardo un hashtag popolare.

4. Come visitatore, voglio poter visualizzare un diagramma a barre che mostra la distribuzione dei tweet in base all'unità di tempo.
   Test: Verificare la variazione dei tweet contenenti "buongiorno" durante il giorno, e dei tweet contenenti "lunedì" durante la settimana.

5. Come visitatore, voglio visualizzare il sito e sfruttarne le funzionalità anche da cellulare.
   Test: visualizzare il sito e fare una ricerca da cellulare.

6. Come visitatore, voglio che eventuali errori vengano gestiti opportunamente e che vengano eventualmente esplicitati tramite un messaggio su schermo.
   Test: Eseguire una ricerca quando il servizio backend è momentaneamente disponibile per far avvenire un errore.

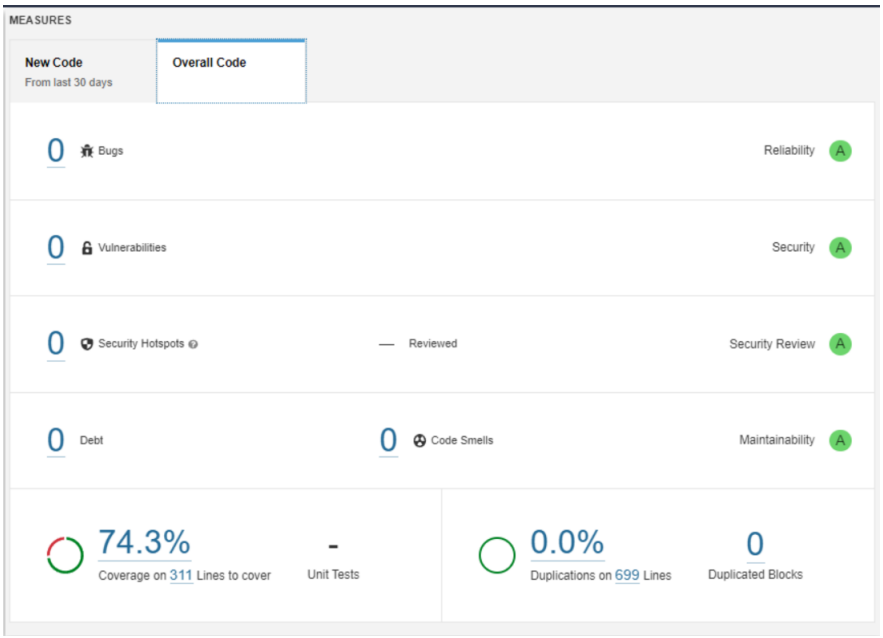

Figura 8: Report di Sonarqube al tempo dello sprint 2.

4.4.1 Sonarqube

In questo sprint la copertura è calata leggermente, scendendo sotto la soglia dell'80%, che è l'obbiettivo del team per la fine del progetto. Nonostante ciò una serie di test sono stati aggiunti per mantenere un risultato così alto a fronte della quantità di nuovo codice. In questa fase sono state richieste funzionalità dal cliente che esulano da quelle pianificate e nel tentativo di portare a termine tutto quello che si erano prefissati, più le nuove richieste, si è preferito mettere in secondo piano i test.

4.5 Burndown

Lo Scrum Maste ha ritenuto opportuno provare a tenere traccia dei progressi dello sprint. Questo è quello che se ne è ricavato:

- 03/11/2022:
  Il PO ha radunato il team per informarlo delle US da svolgere in questo sprint. Il team le ha votate ed è avvenuta la divisione delle mansioni da portare a termine.

- 08/11/2022 Aggiornamenti:
  - si era iniziato a lavorare al grafico a torta;
  - la Word Cloud era quasi conclusa. La developer Andreea ha creato e proposto al team un mock-up per la pagina dell'eredità. Questo è stato

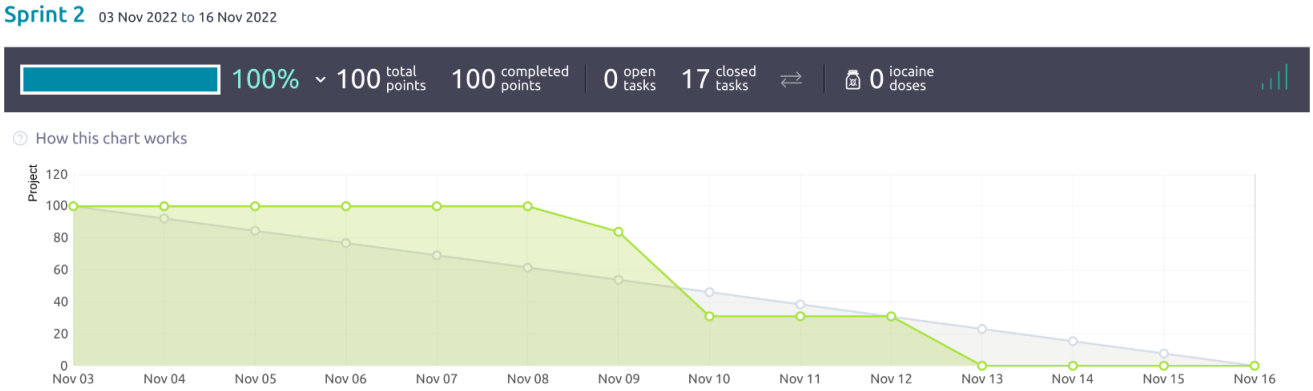

Figura 9: Brundown dello sprint 2.

spedito via email allo stakeholder per approvazione, come richiesta. In attesa di una risposta, tutti gli sviluppatori si sono confrontati per decidere come continuare al meglio le proprie task decidendo: le librerie da usare per i grafici e i sentimenti da usare per la Sentiment Analysis.

- 09/11/2022 Il professor Ciancarini ha suggerito al team un ricevimento per discutere della proposta fatta. Dopo svariati dibattiti e ricerche causati del fatto che il team non aveva trovato un modo per ottenere la parola della ghigliottina prima della trasmissione, si è giunti alla conclusione di provare a ricercare una soluzione creativa realizzabile con gli strumenti a disposizione. Dunque, in seguito ad una serie di analisi per capire in che modo individuare la parola corretta della ghigliottina, il team ha modificato le US in modo da rimanere in linea con le richieste del cliente.

- 14/11/2022 Durante ognuno dei giorni passati il team si è incontrato per aggiornarsi e lavorare insieme a piccoli gruppi e, alternandosi tra di loro, i ragazzi hanno fatto pair programming, sia per velocizzare il processo di scrittura del codice che quello di controllo. Sì è così riuscito a terminare tutte le User Stories, permettendo di anticipare la retrospettiva al giorno seguente.

4.6 Retrospettiva

Sullo **Sprint goal** è riportato il commento: *" Il team si è ritrovato il grosso del lavoro nell'ultima settimana dello sprint. Nonostante ciò sono riusciti a fare tutto."*. Questo è dovuto al ritardo causato dalla comunicazione con lo stakeholder, che ha rallentato il processo di stesura del codice. Malgrado tutto si ritengono soddisfatti, perché con la giusta collaborazione sono riusciti a concludere le US.

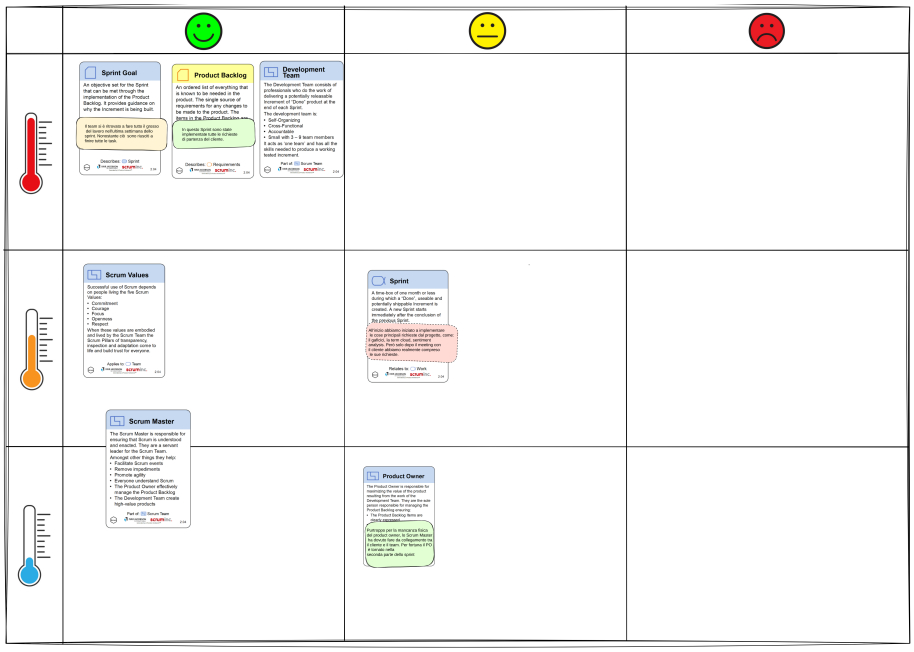

Figura 10: Retrospettiva dello sprint 2.

Product Backlog è stata considerata come molto importante e il team si ritiene appagato perché in questo sprint sono state implementate tutte le richieste di partenza del cliente. Il team ritiene di aver lavorato bene e nell'attesa della risposta del cliente il team ha deciso di portarsi avanti con il da farsi lavorando ai grafici e al Sentiment Analysis.
Un punto fondamentale di questo sprint sarebbe stata la comunicazione tra PO e stakeholder. A causa della mancanza fisica del PO si è dovuto trovare un sostituto. Lo Scrum Mater ha preso momentaneamente le sue veci, facendo del suo meglio, ovviamente grazie anche al sostegno del team e del continuo riscontro telematico con il product owner e il developer team.

4.7 Considerazioni finali

I clienti non sanno quello che stanno cercando finché non lo vedono.

Legge di Humphrey

Al team è stata fatta notare una possibile insicurezza a fronte dei loro svariati tentativi di comunicare con il cliente. La loro non era un'insicurezza, bensì una sicurezza, perchè conoscevano le loro limitate capacità e gli strumenti che avevano a disposizione. L'intento era quello di mostrare allo stakeholder il lavoro che realmente sarebbero riusciti a portare a termine, utilizzando dei mock-up il gruppo cercava di trasmettere la propria visione al cliente. Ovviamente il

cliente non è malleabile così facilmente, per questo si è arrivati a compromessi come non mostrare una classifica live di chi indovinava la parola, ma mostrare una classifica solamente dopo la fine della trasmissione.

5 Sprint 3

Data inizio: 17 Novembre 2022
Data fine: 2 Dicembre 2022

In questo sprint si è decido di dedicarsi principalmente alla pagina degli scacchi. Essa richiedeva un nuovo layout, dei nuovi componenti e una logica di gioco. È in questo sprint che si sono veramente notate le diverse competenze dei membri del team di sviluppo.
Relativamente alla pagina della ghigliottina si è considerato di migliorare l'esperienza dell'utente aggiungendo un podio ed evidenziando i tweet contenenti la parola giusta del gioco.

5.1 Sprint Goal

Lo scopo prefissato era quello di riuscire a impostare una pagina in cui l'utente poteva avviare una partita sul nostro sito e giocare con la comunità di Twitter. Il giocatore doveva poter selezionare la durata dei turni. Ogni volta che l'utente farà una mossa la situazione della partita verrà pubblicata su Twitter e tutti i coloro che vorranno partecipare potranno ri-tweettare la mossa che più ritengono opportuna. La mossa più volte postata sarà giocata se, ovviamente, consona rispetto alle regole del gioco.

5.2 Sprint backlog

In seguito al primo incontro del team in cui si è svolto un brainstorming riguardo a come sviluppare la pagina degli scacchi, il PO ha stilato le seguenti User Stories:

1. Visualizzare la scacchiera;
2. Classifica ghigliottina;
3. Visualizzare tentativi ghigliottina;
4. Durata del turno a scelta;
5. Doppio Timer;
6. Giocare la partita di scacchi;
7. Layout della pagina degli scacchi;
8. Grafico a barre della ghigliottina.

5.3 Definition of Done

A causa dell'eccessivo carico di lavoro e del tempo necessario per documentarsi su esso, il team non sarebbe stato in grado di fare i test. Dunque, dall'usuale Definition of Done, utilizzata fino a quel punto del progetto, hanno ritenuto opportuno rimuovere la parte relativa al testing.

5.4 Test

Nonostante la decisione di non testare in questo sprint il codice, si è comunque dovuto testare il funzionamento corretto del User Stories implementate. Sottostante possiamo trovare il resoconto delle US e la loro verifica.

1. Come visitatore, voglio visualizzare la scacchiera che mostra lo stato della partita di scacchi giocata tramite tweet.
   Test: osservare che la scacchiera si della dimensione giusta per essere piacevole alla vista.

2. Come visitatore, voglio visualizzare una classifica che mostri in ordine i primi ad indovinare la parola del giorno.
   Test: osservare i risultati della puntata giornaliera.

3. Come visitatore, voglio una pagina web che mostri i tweet di chi ha provato ad indovinare la parola del giorno, evidenziando chi ha indovinato e chi ha sbagliato.
   Test: osservare i risultati di una puntata recente.

4. Come giocatore di scacchi, voglio poter scegliere quanto dureranno i turni della partita, prima di avviarla.
   Test: giocare una partita con turni da 5 minuti e da un'ora.

5. Come giocatore di scacchi, voglio visualizzare due timer insieme allo stato della partita: uno che mostri il tempo rimanente nel turno dell'avversario, e uno che faccia lo stesso durante il mio turno.
   Test: Giocare una partita e aspettare di perdere a tavolino.

6. Come visitatore del sito, voglio la possibilità di avviare e giocare una partita di scacchi in cui le mosse del mio avversario saranno decise tramite "voto" su Twitter.
   Test: avviare e completare una partita.

8 Come visitatore voglio visualizzare, in base ad una puntata, un grafico a barre che mostri la quantità di tentativi distribuiti nel tempo, facendo differenza tra chi ha indovinato e chi no.
   Test: osservare i risultati della puntata giornaliera.

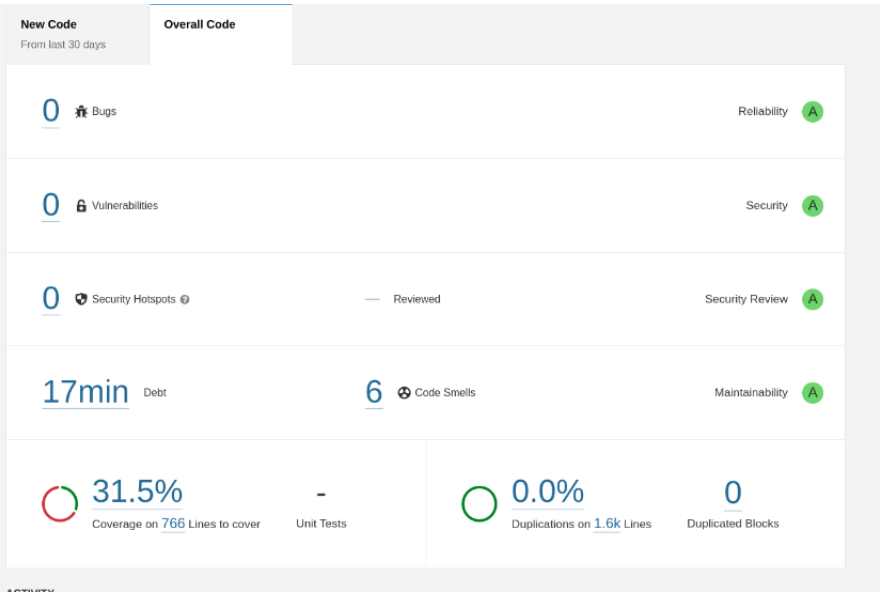

Figura 11: Sonarqube dello sprint 3.

5.4.1 Sonarqube

Rispetto al secondo sprint, (vedi Fig. 8) lo score è peggiorato drasticamente. Trascurare i test è stata una decisione dell'intero Team che ha preferito concentrarsi maggiormente sull'implementazione della pagina di Scacchi. Comunque consapevoli del fatto che i test sono parte integrante dello sviluppo software, hanno deciso di lavorarci nello sprint successivo.

5.5 Burndown

Perfino dal grafico Burndown 12 possiamo notare come le difficoltà riscontrate abbiano rallentato il processo di sviluppo e di stesura del codice. Infatti, a differenza di altre volte in cui la prima fase era di informazione e la seconda di sviluppo, in questo sprint sono risultate più rare le volte in cui si riusciva a completare il fabbisogno giornaliero, ciò probabilmente a causa dei Daily Scrum.

5.6 Retrospettiva

Nella retrospettiva 16 di questo sprint si è cercato di capire a cosa fossero dovute le difficoltà riscontrate, in modo da cercare di migliorare la comunicazione tra i componenti. Notiamo che sulle Essence Card **Cross-Function Team** e **Self Organization** sono riportati degli appunti riguardanti i principali intoppi riscontrati quali le competenze personali e l'organizzazione. Infatti, non tutti avevano le *skill* per sviluppare il gioco, ad esempio non tutti sanno giocare a

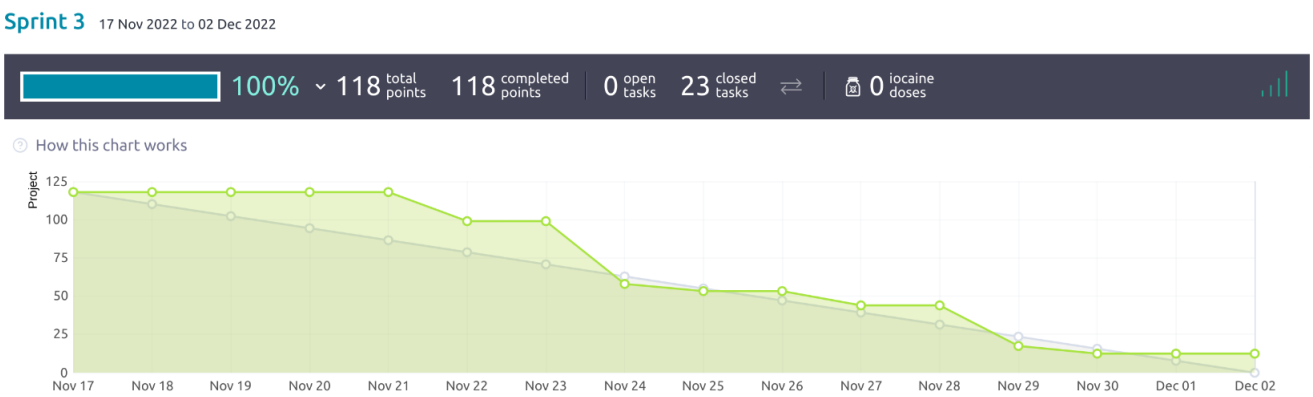

Figura 12: Burndow dello sprint 3.

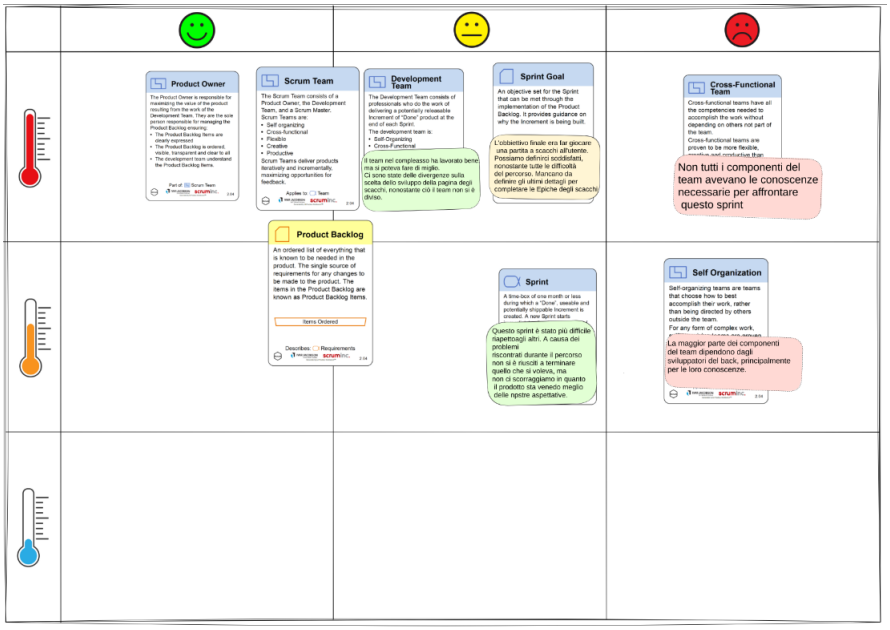

Figura 13: Retrospettiva dello sprint 3.

scacchi. Per questo motivo, durante la fase lavorativa, avevano un continuo bisogno di consultarsi con gli sviluppatori esperti, rallentando la loro produzione e quella di coloro che provavano ad aiutarli. Documentarsi richiedeva più tempo di quello a disposizione, perciò ogni qualvolta fosse possibile si è deciso di fare pair programming.
Per quanto riguarda lo **Sprint Goal** lo Scrum Team si ritiene soddisfatto, considerata la difficoltà dello sprint.
La carta dello **Sprint** riporta come le difficoltà riscontrate abbiano portato ad un po' di debito tecnico.

5.7 Considerazioni finali

Per questo sprint non ci sono particolari considerazioni da riportare. Nella Review il cliente ha potuto testare il prodotto, e non ha richiesto nuove funzionalità.
I ragazzi hanno un piccolo debito tecnico da recuperare, ma non dovrebbe corrompere i loro obbiettivi del prossimo sprint. Si sono ripromessi di impegnarsi al massimo per finire il sito entro la fine dello sprint 4, cosi da rispettare le scadenze prefissate dal cliente.

6 Sprint 4

Data inizio: 4 Dicembre 2022
Data fine: 18 Dicembre 2022

Lo sprint è iniziato con un imprevisto. Lo stakeholder Missiroli ha deciso di fare una richiesta a tutti i gruppi e sostituire la pagina relativa a "Reazione a catena", che il gruppo aveva momentaneamente trascurato a causa della sua inattività nel periodo dato, con l'emissione **Propaganda**, in particolare con il suo rinomato gioco **Fantacitorio**. Questo inconveniente ha causato dubbi e incertezze nel team soprattutto a causa del tempo ridotto e della eccessiva mole di lavoro da portare a termine.

6.1 Sprint Goal

Questa volta ci sono stati 3 sprint goal:

- Finire la logica degli scacchi;
- Fare la pagina di Fantacitorio;
- Testare il più possibile il codice.

6.2 Sprint backlog

Le US di questo sprint sono:

1. Visualizzare punteggi politici;

2. Classifica;
3. Vittoria scacchi;
4. Resa scacchi;
5. Squadre Fantacitorio;
6. Ricerca squadre;
7. Grafico a barre delle mosse di scacchi.

La mole di lavoro di queste User Stories è maggiore di quella di qualsiasi altro sprint passato, il che era proprio che ciò che il team voleva evitare. Infatti, avevano cercato di portare a termine la maggior parte del lavoro negli sprint precedenti, in modo che nell'ultimo potessero dedicarsi a perfezionare i servizi esistenti, ripulire il codice e testarlo.
L'intoppo ha portato il team a riflettere e concludere che il mondo del lavoro è imprevedibile e che nei propri piani bisogna sempre tenere in considerazione gli imprevisti, lasciando quindi un margine di tempo per gli eventi non programmati.

6.3 Definition of Done

Per questo sprint si torna alla Definition of Done iniziale, quella che si è usata per tutti gli sprint tranne che per il terzo. Si è in particolare deciso di procedere con il pair programming. In questo modo il team contava di essere il quanto più veloce e preciso possibile nella stesura del codice. Per evitare, inoltre, qualsiasi tipo di problema, prima di ritenere conclusa una US, il codice veniva analizzato ed eventualmente approvato da almeno un altro componente del team.

6.4 Test

Le US di questo sprint sono state più del previsto e molto diverse da quelle che inizialmente dovevano essere. Di seguito riportiamo la descrizione delle US scelte:

1. Come visitatore, vorrei visualizzare in una sezione della pagina una classifica dei politici con più punti.
   Test: Verificare che la classifica corrisponda a quella nel file redatto da '@RosyilCapo'.
2. Come giocatore di scacchi, voglio che una volta terminata la partita venga mostrata una schermata e pubblicato un tweet con l'esito.
   Test: Verificare che la schermata venga pubblicata su Twitter.
3. Come giocatore di scacchi, voglio potermi arrendere durante la partita con un apposito pulsante.
   Test: Giocare una partita e arrendersi verificando che il tutto abbia un comportamento corretto.

4. Come giocatore del Fantacitorio, voglio poter visualizzare le squadre condivise dall'account ufficiale e il nome utente di chi le ha pubblicate.
   Test: Osservare le squadre attuali.

5. Come giocatore del Fantacitorio, voglio poter cercare fra le immagini delle squadre per vedere solo quelle corrispondenti a un certo username.
   Test: Cercare delle squadre precise, cercare se userInesistente ne ha pubblicata una.

6. Come giocatore di scacchi, voglio osservare la popolarità delle mosse giocate tramite Twitter attraverso un grafico a barre.
   Test: giocare una partita e ammirare il grafico, vedere cosa succede quando nessuno fa una mossa.

La scelta del PO ha portato ad eliminare dal progetto altre User Stories come:

- Aumentare il numero di tweet.
  Come visitatore, in caso i risultati fossero troppi per entrare in una sola pagina, vorrei avere la possibilita' di visualizzare piu' tweet corrispondenti alla mia ricerca grazie ad un apposito selettore o pulsante.

- Ghigliottina prima della parola del giorno.
  Come visitatore, voglio visualizzare grafici che mostrano le soluzioni più twittate aggiornati "in tempo reale" durante la puntata prima che venga rivelata la parola del giorno.

- Hall of fame ghigliottina.
  Come visitatore, voglio poter visualizzare una classifica stilata in base a quanti giorni di fila le persone sono riuscite ad indovinare la parola giusta.
  Test: Visualizzare la classifica, e provare a scalarla.

Queste funzionalità sono quelle che al team sarebbe piaciuto completare se non ci fosse stata la epica del Fantacitorio.

6.4.1 Sonarqube

Nell'ultimo sprint, come da programma, si è aumentato la copertura dei test fino al 90%, che è oltre la soglia dell'80% che ci eravamo prefissati. Il team ha inoltre aggiunto test per le nuove funzionalità chieste dai clienti all'ultimo momento, per rimanere in linea con il nostro obbiettivo. Il testing non ha portato a grandi sorprese, in quanto il codice non testato scritto in precedenza era stato comunque validato a mano a fondo, in preparazione alla periodica demo bi-settimanale.

La copertura finale è buona, ma non perfetta per un motivo intrinseco al conteggio della copertura nel linguaggio Go. A differenza di altri linguaggi, come Java o C#, dove si usano le eccezioni per passare gli errori, Go preferisce restituire un errore come un vero e proprio valore. Dunque, la gestione delle "eccezioni" avviene tramite `if`-statement e non tramite appositi `try-catch`. Per ottenere una copertura perfetta dunque, sarebbe necessario creare casistiche in cui si

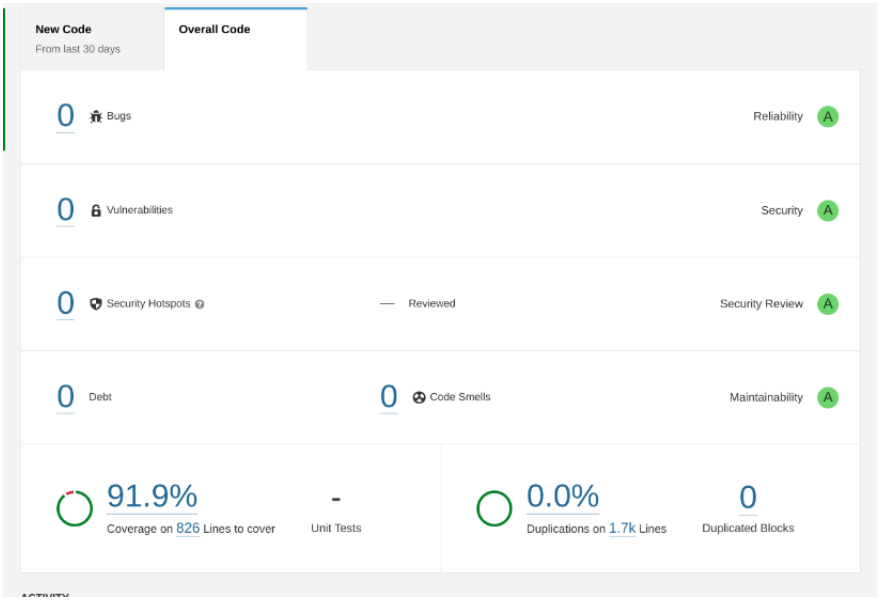

Figura 14: Sonarqube dello sprint 4.

esplorano tutti i possibili errori, mentre in altri linguaggi basterebbe generare una delle tante casistiche coperte da un `catch`. Per questo motivo, le percentuali della copertura in Go risultano sì basse se paragonate a quelle ottenute in altri linguaggi, ma sono anche più realistiche e consentono tramite strumenti come SonarQube di vedere a colpo d'occhio quali casistiche d'errore non vengono controllate.

6.5 Burndown

Il developer Catalini ha recuperato il debito tecnico prima del primo Daily Scrum (il quale solitamente si svolge per analizzare i requisiti), ciò ha permesso di iniziare a lavorare alle nuove US senza pensare alle vecchie.

- Il 5/12/2022: Primo Daily Scrum in cui si è analizzato il gioco di Fantacitorio, andando innanzitutto ad analizzare le risorse che si possedevano, come i tweet della pagina ufficiale, le immagini pubblicate dai fan, il foglio excel della raccolta dei punteggi e la lista dei politi in gara.
  Il punto principale da esaminare era se il team sarebbe stato in grado di soddisfare tutte le richieste del cliente considerando il materiale a disposizione. Si è giunti alla conclusione di provare a realizzare tutte le richieste garantendosi degli aggiornamenti reciproci, così che in caso di difficoltà si potesse contattare il prima possibile lo stakeholder.

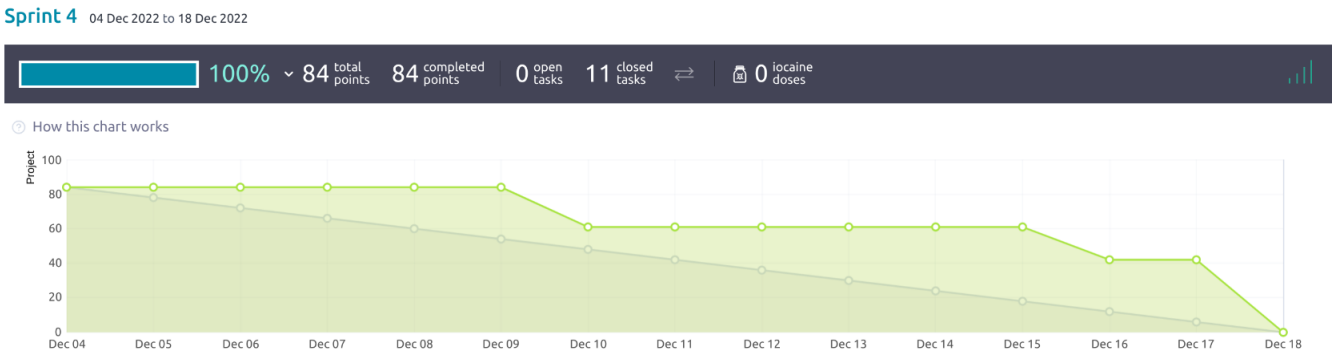

Figura 15: Burndown dello sprint 4.

- Il 8/12/2022: Quello che sarebbe dovuto essere un Daily Scrum, si è rivelato essere un meeting in cui i ragazzi hanno programmato in compagnia, cercando di velocizzare il processo di sviluppo.
- Il 10/12/2022: Alcuni membri del Gruppo di sviluppo si sono incontrati per andare avanti con la pagina del Fantacitorio. I sviluppatori Yonas e Gaia, si sono occupati del front-end, invece Gianmaria del back-end. Coloro che non potevano esserci di persenza hanno lavorato in totale autonomia da casa. A fine giornata hanno aggiornato lo Scrum Mater su quello che sono riusciti a completare.
- Il 17/12/2022: Il team non ha potuto incontrarsi di persona, ma ha comunque fatto un Daily Scrum su Mattermost.
  L'argomento principale è stato relativi ai test. Con i test della parte di scacchi si era raggiunto il coverage del 60, 83%, ma ovviamente si sapeva che bisognava migliorare prendendo in considerazione tutte le altre parti. Infatti a fine giornata i ragazzi hanno dato il massimo raggiungendo il 91,9%.

6.6 Retrospettiva

Sprint Planning sulla carta è riportato il seguente commento: *"A causa dell'imprevisto da parte dello stakeholder si è dovuto rivedere lo Sprint Planning precedentemente fatto alla fine dello sprint scorso a causa del debito tecnico. Si sono dovute rivedere alcune US, ma soprattutto quello che sarebbe stato in grado di fare il Team di sviluppo nel poco tempo a disposizione. Alla fine non si è riuscito a fare tutto, ma lo Scrum Team si ritiene soddisfatto."* Collegato a questo c'è anche il commento della carta del **Product Backlog**: *"Purtroppo non si sono riuscite a introdurre nell'ultimo sprint tutte le User Stories del Backlog, questo a causa dell'imprevisto del Fantacitorio."* Sulla carta dell'**Increment**, che fino ad ora non si era usata viene ribadito il fatto che non si è riuscito ad

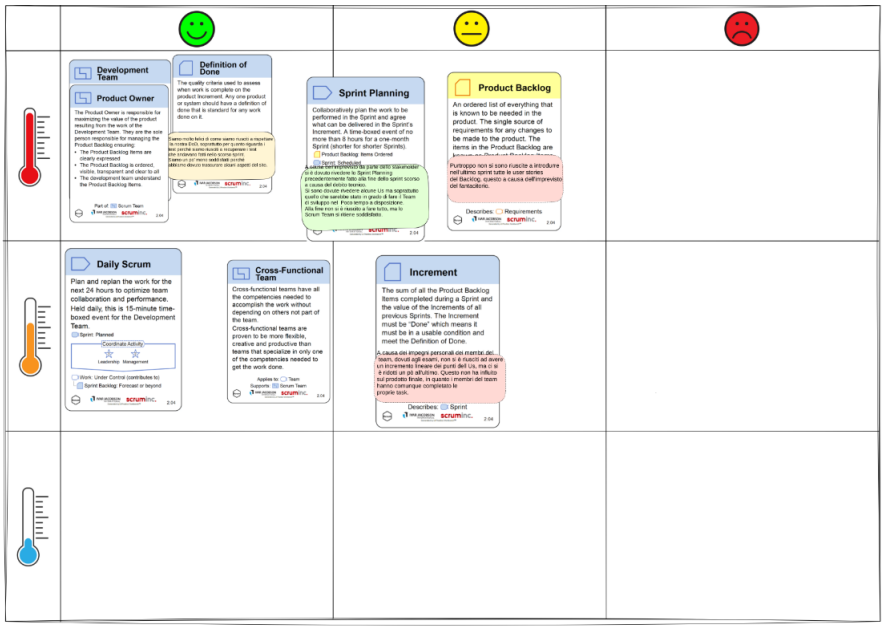

Figura 16: Retrospettiva dello sprint 4.

avere un incremento costante, perché a differenza degli altri sprint ogni componente ha iniziato a prepararsi per gli esami. L'ultima carta con un commento è quella della **Definition of Done**, in quanto il team si ritiene molto soddisfatti dei test fatti.

6.7 Considerazioni finali

Questo ultimo sprint non è stato facile, per diversi aspetti. Il primo è dovuto appunto all'imprevisto dello stakeholder Missiroli. Secondo lo Scrum Team aveva organizzato anticipante lo sprint planning, infatti si pensava di revisionare il sito completamente, aggiungendo le funzionalità mancanti, gli ultimi grafici e sopratutto concentrarsi sui test. Era pianificato come sprint più leggero, perché il team aveva già previsto la difficoltà del periodo, causata dai imminenti esami, il che ha reso gli sviluppatori meno disponibili. Purtroppo è successo proprio il contrario di quello che si voleva, lo sprint era saturo di lavoro da portare a termine, nonostante il PO abbia cercato di alleggerirlo il più possibile. Malgrado ciò il Developer team ha dato il meglio di se finendo le US scelte.

7 Processo di sviluppo

Come si può notare per sviluppare il progetto è stata usata la metodologia Agile. Essa richiede tre artefatti, che sono: **Product Backlog**, **Sprint Backlog** e **Increment**. I quali a loro volta sono composti da: Product Goal, Sprint Goal e Definition of Done. Il primo, che fa parte del Product Backlog, descrive uno stato futuro del prodotto e viene utilizzato come obiettivo a lungo termine. Diversamente lo Sprint Goal descrive un obiettivo più a breve termine. La Definition of Done, parte dell'increment, descrive un incremento di software, e definisce quando un certo lavoro deve ritenersi completato garantendo quindi il rilascio di un prodotto adeguato per l'utente.
Tutti questi strumenti sono stati utilizzati dallo Scrum Team, cercando di rendere il più realistico possibile un processo educativo ad un lavoro reale, per questo motivo hanno utilizzato, Sprint Planning, Daily Scrum, la Review e la Retrospettiva. Il team ha cercato di sfruttare al meglio questi strumenti.
Per essere in linea con i valori di Scrum, è stata importante anche la suddivisione dei ruoli all'interno del Team. Quando le persone di un Team sono ben preparate e in gamba, allora ha senso utilizzare il pieno potenziale di ognuno, condividendo le responsabilità anziché farlo dirigere da una singola persona, tale è giovevole per evitare colli di bottiglia.
Ma il risultato più importante di questo modo di lavorare è un maggior coinvolgimento e una maggior partecipazione di tutti i membri del team, questo porta indubbiamente ad un aumento della produttività e del benessere di ognuno.

7.1 Strumenti utilizzati

È stato anticipato, già nel paragrafo dedicato all'introduzione dello sprint 1, quelli che sono stati gli strumenti utilizzati. Analizzeremo le modalità di utilizzo di questi da parte del team e come abbiano migliorato l'esperienza di sviluppo.

Gitlab è stato lo strumento principale per lo sviluppo del software. Essendo una piattaforma web open source alcuni ragazzi ne avevano già fatto uso per altri progetti. Anche chi non l'aveva mai utilizzato non ha avuto particolare difficoltà nell'imparare ad usarlo: l'esperienza pregressa con piattaforme simili come Github ha fatto la differenza. Il team era solito utilizzare tutti tool che la piattaforma offriva come le *branch*. Per ogni US nuova si apriva una branch, per evitare così che più persone lavorassero contemporaneamente allo stesso codice. Quando una User Story veniva assegnata a più di una persona si cercava di fare pair programming. Ogni volta che si apriva una nuova branch si doveva aprire anche una *Merge requests* essenziale per quando successivamente si sarebbe dovuto unire il codice nella branch principale, il *main*.

Per la messa in produzione abbiamo utilizzato *Docker*. Siamo stati in grado di progettare una pipeline con i Runner di GitLab per costruire una release del nostro software ad ogni tag di git in modo automatico. Questa release è rilasciata sotto forma di immagine Docker e viene caricata in automatico sullo storage GitLab Registry per un facile accesso da parte dell'amministratore di

sistema che ne gestirà il deployment.

Mattermost è stata una piattaforma nuova per tutti. Nessun corso prima di questo aveva richiesto di prendere famigliarità con questo tool ma anche questa volta non è stato complicato adattarsi. La possibilità di avere diversi canali facilita la comunicazione tra i componenti del team di sviluppo, nondimeno quella tra SM o PO e il team. In questo caso i canali reputati necessari sono stati 5:

1. **DEV**: utilizzata principalmente dal team di sviluppo per scambiarsi consigli, comunicare i propri progressi e in generale discutere di quello che comprendeva il codice.

2. **Epiche ed User Stories**: principalmente serviva al Product Owner per comunicarci, quando non era possibile farlo di persona, le User Stories nuove. Spesso veniva utilizzata anche dagli altri ragazzi per consigliargli future US da redarre.

3. **Information**: canale utilizzato per gli avvisi importanti. Sfruttato a volte dallo Scrum Master per motivare e ricordare le cose da fare.

4. **Meeting**: dedicato a prefissare gli incontri settimanali, o fare chiamate quando non ci si poteva incontrare di persona.

5. **Off-Topic**: per tutte le conversazioni che non riguardavano il progetto.

Essendoci un membro del team impossibilitato ad esserci in presenza a causa di un percorso di studi all'estero, Mattermost è risultato anche estremamente utile per mantenere un ottimo contatto con lui.

Taiga è stata la piattaforma più utile di tutte. Consigliata per tenere traccia del progressi del progetto. La dashboard è intuitiva e consente ai collaboratori di gestire e completare le loro attività con assoluta facilità. Completamente compatibile con il modello Agile ha le sezioni dedicate per le epiche, il backlog del progetto e quelle per ogni sprint. Molto efficiente dal punto di vista gestionale in cui permette di assegnare task a membri del team ed è stata utilizzata molto per scandire le scadenze delle User Stories. Oltre tutto ha anche un apposito spazio in cui è possibile caricare il materiale come mock-up, retrospettive e documentazioni del lavoro, che può risultare veramente utile in alcuni casi.

Come ultimo strumento si ha **Sonarqube** che consente al team di scrivere codice più pulito e più sicuro. Esso garantisce un'ispezione continua del codice e fornisce fornisce protezione al progetto esaminato.

7.2 Analisi temporale

Il grafico in figura 17 mostra quante ore ogni membro del team ha dedicato a ciascun sprint. All'incirca ognuno ha spesso in media 20 ore per ogni sprint.

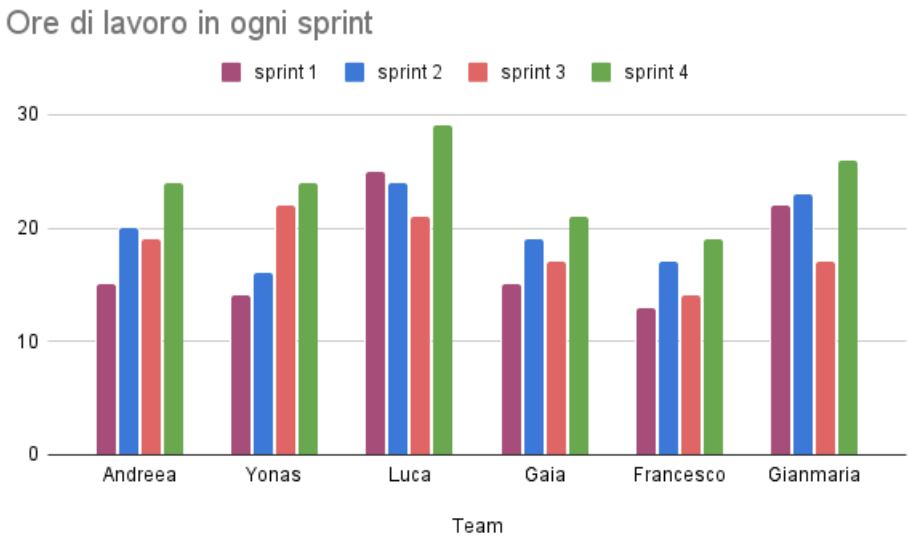

Figura 17: Grafico che indica quante ore ha lavorato ciascun membro in ogni sprint.

Ovviamente questa è un'approssimazione, in quanto non c'è stato modo o necessità di contarle esattamente Non si è infatti utilizzato nessun software per conteggio del tempo; lo SM ha optato per un metodo tradizionale: su un foglio excel ognuno teneva conto delle ore che aveva svolto in un determinato giorno andando a commentarle brevemente per tenere traccia del lavoro svolto.

7.3 Prodotto Finale

Ecco qui i link al sito web della demo.
Il link Taiga.
Il link di Sonarqube.
Il link di Mattermost.
Il link di Gitlab.