

Progetto di Ingegneria del Software: Team3

Juan Guillermo Jaramillo Saa (SM)
juan.jaramillosaa@studio.unibo.it

Daniele Tarek Iaisy (PO)
danieletarek.iaisy@studio.unibo.it

Alexandru Bogdan Nicolescu (DEV)
alexandru.nicolescu@studio.unibo.it

Riccardo Fava (DEV)
riccardo.fava5@studio.unibo.it

Andrea Largura (DEV)
andrea.largura@studio.unibo.it

14 Dicembre 2022

Contents

1	Prodotto	4
1.1	Descrizione del prodotto	4
1.2	Ricerca	4
1.2.1	Filtraggio	4
1.2.2	Statistiche	4
1.3	Epiche	4
1.4	Backlog di prodotto	5
2	Diagrammi	6
2.1	Caso d'uso iniziale	7
2.2	Caso d'uso (aggiunto)	8
2.3	Diagramma di Deployment	8
2.4	Continuous Integration, Continuous Deployment	8
3	Sprint	10
3.1	Sprint 0	10
3.1.1	Team building	10
3.1.2	Scrumble	10
3.1.3	Scheda di autovalutazione	11
3.1.4	Flusso di lavoro	12
3.2	Sprint 1	13
3.2.1	Sprint goal	13
3.2.2	Sprint backlog	13
3.2.3	Definition of Done	13
3.2.4	Burndown dello sprint	13
3.2.5	Test	13
3.2.6	Scelte progettuali	13
3.2.7	Sprint Review	14
3.2.8	Retrospettiva	14
3.3	Sprint 2	15
3.3.1	Sprint goal	15
3.3.2	Sprint backlog	15
3.3.3	Definition of Done	16
3.3.4	Burndown dello sprint	16
3.3.5	Test	17
3.3.6	Scelte progettuali	17
3.3.7	Sprint Review	17
3.3.8	Retrospettiva	18
3.4	Sprint 3	19
3.4.1	Sprint goal	19
3.4.2	Sprint backlog	19
3.4.3	Definition of Done	20
3.4.4	Burndown dello sprint	20
3.4.5	Test	20

3.4.6	Scelte progettuali	20
3.4.7	Pipeline	21
3.4.8	Sprint Review	21
3.4.9	Retrospettiva	22
3.5	Sprint 4	22
3.5.1	Sprint goal	22
3.5.2	Sprint backlog	23
3.5.3	Definition of Done	23
3.5.4	Burndown dello sprint	23
3.5.5	Test	23
3.5.6	Scelte progettuali	24
3.5.7	Retrospettiva	24
3.6	Alcuni dati sul processo	26
3.6.1	Sonarqube	26
3.6.2	Gitinspector	27
3.6.3	Burndown finale	27
4	Artefatti	28

1 Prodotto

1.1 Descrizione del prodotto

Si vuole costruire un prodotto software capace di raccolta e analisi dei tweet di Twitter.

I tweet possono riferirsi ad una persona, ad un luogo, ad un evento, ad un gioco, ad una trasmissione televisiva.

Si vogliono utilizzare i tweet anche per giocare in gruppo o supportare giochi televisivi.

Le sezioni a seguire riportano le funzionalità che si vogliono implementare.

1.2 Ricerca

- Per parola chiave: vengono restituiti i tweet contenenti una data parola chiave
- Username: vengono restituiti i tweet pubblicati da un dato utente
- Hashtag: vengono restituiti i tweet per un dato hashtag

È inoltre possibile scegliere se attivare lo streaming dei tweet in tempo reale.

1.2.1 Filtraggio

- Per data: è possibile scegliere gli estremi dell'arco temporale preferito (all'interno dell'ultima settimana)
- Per numero di risultati: è possibile scegliere il numero di risultati da visualizzare

1.2.2 Statistiche

- Sentiment analysis: ogni tweet riceve un punteggio (Buono, Neutro, Male).
- Sentiment analysis della ricerca: ogni ricerca genera un diagramma con parole positive, negative e neutre utilizzate nell'intera ricerca.
- Wordcloud: diagramma delle parole più ricorrenti all'interno della ricerca.

1.3 Epiche

- L'eredità: deve essere possibile raccogliere i tweet di chi prova a indovinare la ghigliottina, per visualizzare (in ordine temporale, o su una mappa) tutti coloro che indovinano
- Scacchi: deve essere possibile poter giocare a scacchi in modo tale che le mosse avversarie siano scelte per maggioranza

- Fantacitorio: deve essere possibile raccogliere i tweet delle squadre del fantacitorio o di un utente in particolare e anche visualizzare una classifica con politici e punti (e il miglior punteggio singolare di tutta la classifica).

1.4 Backlog di prodotto

- **US:** Homepage

Descrizione: Come visitatore di pagine web vorrei poter visualizzare la homepage dell'applicazione

Punti: 8

- **US:** Ricerca per parola chiave

Descrizione: Come abitante di una bolla, voglio filtrare i tweet per parola chiave, così da vedere solo quello che interessa a me

Punti: 13

- **US:** Ricerca per utente

Descrizione: Come genitore apprensivo vorrei un modo per cercare i tweet di mio figlio per monitorarlo

Punti: 15

- **US:** Visualizzazione diagramma a barre

Descrizione: Come analista vorrei avere un grafico che mi permette di vedere il numero di tweet raccolti nel tempo, così da sapere gli andamenti di popolarità dei topic

Punti: 13

- **US:** Visualizzazione tweet su mappa

Descrizione: Come genitore apprensivo vorrei un modo per cercare i tweet di mio figlio e visualizzarli sulla mappa

Punti: 13

- **US:** Visualizzazione sentiment analysis

Descrizione: Come analista di dati vorrei un grafico a torta sul sentiment dei tweet, per farmi un'idea di cosa ne pensino gli utenti

Punti: 28

- **US:** Ricerca per data

Descrizione: Come storico, voglio poter scegliere un intervallo di date, per selezionare i tweet del solo periodo che sto studiando

Punti: 14

- **US:** Scelta del numero dei risultati

Descrizione: Come binge tweeter, voglio poter visualizzare piu' di 10 tweet alla volta

Punti: 15

- **US:** Visualizzazione term cloud

Descrizione: Come membro dell'accademia della crusca vorrei vedere le

parole dei tweet selezionati su una term cloud, così da poter vedere le parole più utilizzate

Punti: 13

- **US:** La ghigliottina

Descrizione: 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

Punti: 32

- **US:** Proposta sfida

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

Punti: 18

- **US:** Mossa sfida

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

Punti: 23

- **US:** Punteggi settimanali

Descrizione: Come spettatore del fantacitorio voglio poter visualizzare la classifica emendabile settimanale di sempre dei politici.

Punti: 26

- **US:** Immagini squadre

Descrizione: Come spettatore del fantacitorio voglio poter sfogliare le immagini delle squadre registrate

Punti: 24

- **US:** Squadra utente

Descrizione: Come spettatore del fantacitorio voglio poter vedere le squadre dei miei amici

Punti: 19

2 Diagrammi

Questi diagrammi sono stati fatti tramite lo strumento open source plantuml <https://plantuml.com/>.

Si tratta di uno strumento che permette di creare diagrammi a partire da testo.

2.1 Caso d'uso iniziale

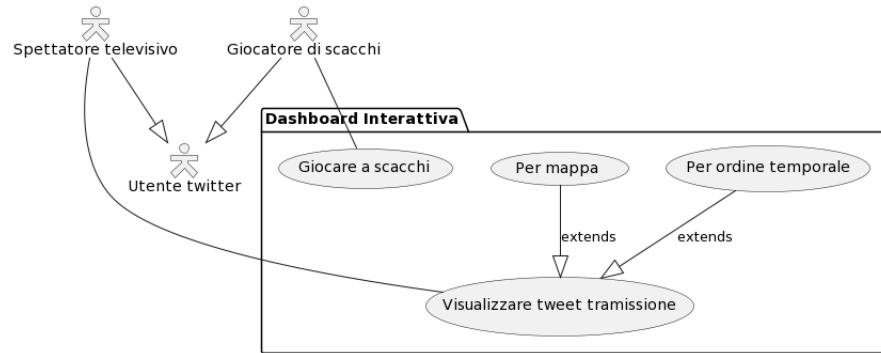


Figure 1: Caso d'uso, parte 1

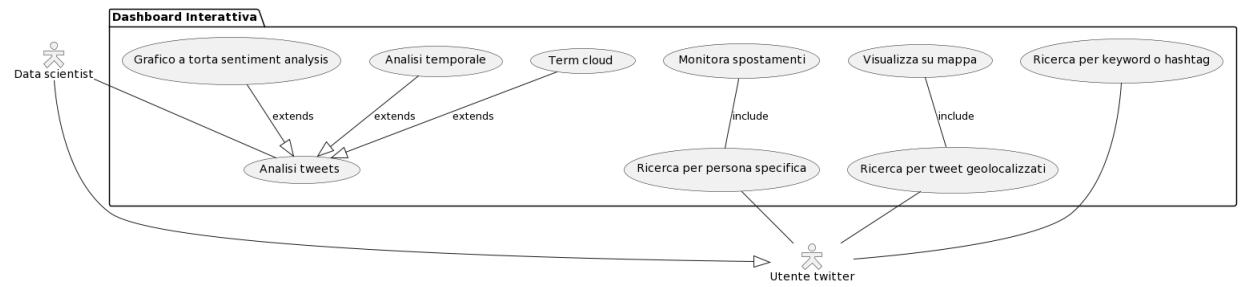


Figure 2: Caso d'uso, parte 2

2.2 Caso d'uso (aggiunto)

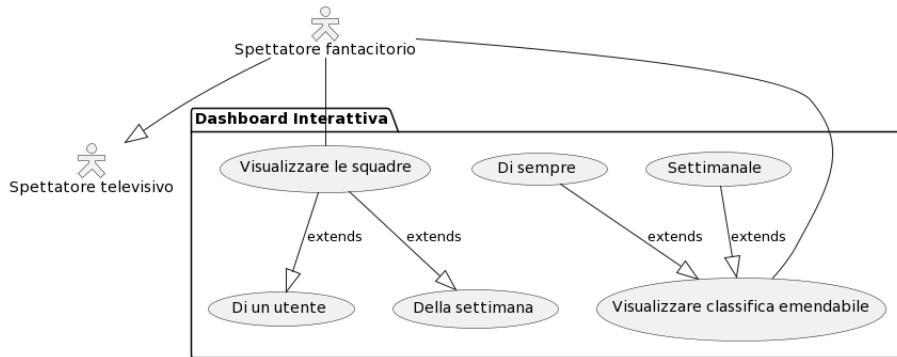


Figure 3: Caso d'uso, parte 3 (aggiunta dello sprint 4)

2.3 Diagramma di Deployment

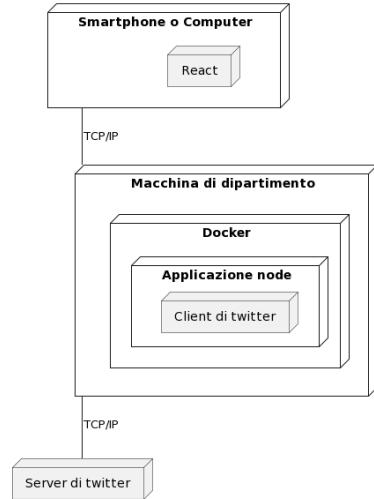


Figure 4: Diagramma di deployment

2.4 Continuous Integration, Continuous Deployment

Quando viene effettuato un push sul main viene effettuata l'integrazione con l'aggiornamento degli item su *Taiga* (se vengono trovati messaggi con TG-#task #staus).

Dopodiché vengono fatti partire gli stage di test, build e deploy automatizzato. In questo modo abbiamo <http://site212236.tw.cs.unibo.it> sempre aggiornato

all'ultima versione.

Visto che abbiamo adottato [GitHub Workflow](#) come flusso di lavoro git, ci è stato utile attivare degli stage ad-hoc per le merge request.

Quando viene effettuata una merge request, viene fatto partire lo stage di test e di build, in questo modo riuscivamo a capire fin da subito se i nostri branch fossero parzialmente pronti ad essere mergiati al main (non prima di essere stati visionati da almeno un altro sviluppatore)

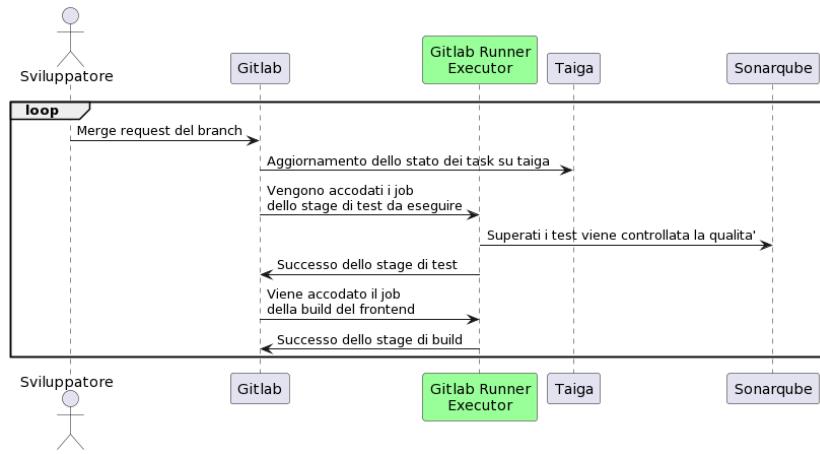


Figure 5: Diagramma di sequenza, merge requests

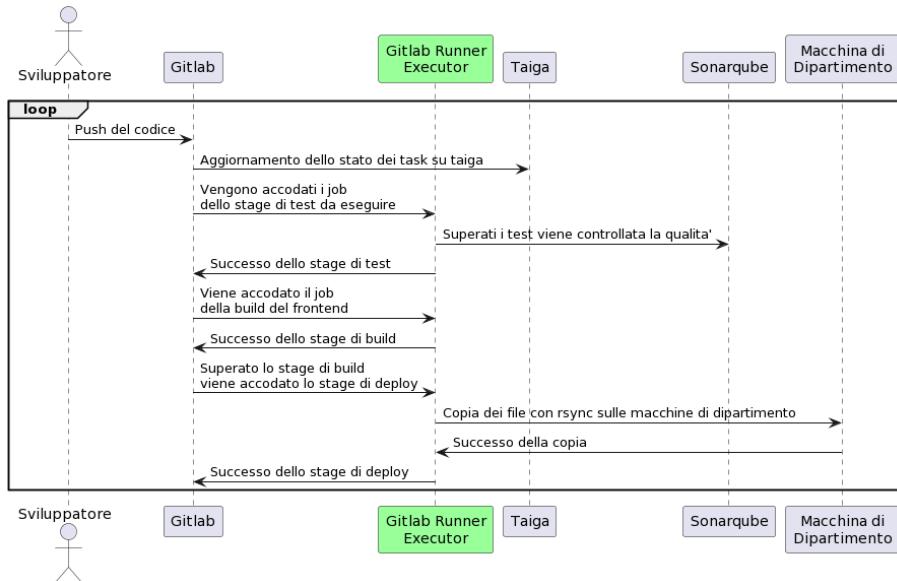


Figure 6: Diagramma di sequenza, main

3 Sprint

3.1 Sprint 0

3.1.1 Team building

Il team è stato creato grazie a Trello, prima di questo progetto solo Juan e Alex si conoscevano.

Per questo motivo abbiamo speso la maggior parte di questo sprint a conoscerci tra di noi.

3.1.2 Scrumble

Abbiamo effettuato la partita di scrumble in modalità mista (tramite *Discord*), con Daniele che ha condiviso lo schermo.

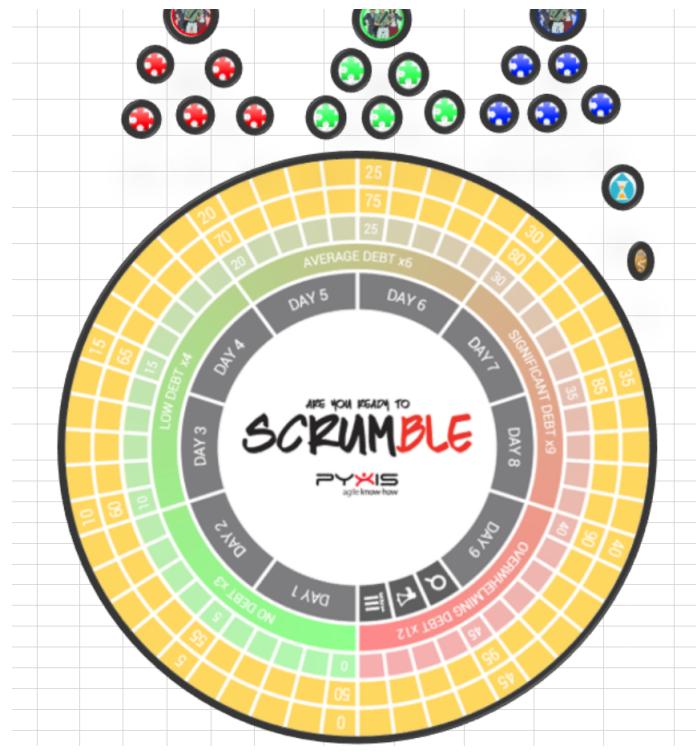


Figure 7: Screenshot della board che abbiamo utilizzato

La partita, in linea generale, è andata bene, siamo riusciti a rompere il ghiaccio per conoscerci meglio e i ruoli intrapresi durante la partita sono rimasti tali per l'intero progetto.

3.1.3 Scheda di autovalutazione

Al termine della partita abbiamo provveduto a compilare la scheda di autovalutazione, riportata a seguire:

GOAL	QUESTIONS	EVALUATION	Danielle PO	Juan SM	Alex	Riccardo	Andrea
Learn	Q1	1 = no idea of the Scrum roles 5 = perfect knowledge of the roles and their jobs	5	5	4	4	4
	Q2	1 = couldn't repeat the game 5 = could play the game as a Scrum Master by himself	5	5	5	5	4
	Q3	1 = totally lost 5 = leads the game driving the other players	5	5	4	4	3
Practice	Q4	1 = feels the game is unrepeatable 5 = feels the game could be played in any situation	2	3	4	4	5
	Q5	1 = 0 to 3 stories 2 = 4 to 6 3 = 7 to 9 4 = 10 to 12 5 = 13 to 15	3	3	3	3	3
	Q6 ONLY DEV TEAM	1 = abnormal difference from the other players 5 = coherent and uniform with the group most of the time			5	4	4
Cooperation	Q7	1 = never speaks with the other players 5 = talks friendly to anyone in every situation	5	4	4	4	4
	Q8	1 = never puts effort in doing something 5 = every time is willing to understand what is going on	5	5	5	5	5
	Q9	1 = never asks for an opinion 5 = wants to discuss about every topic	4	4	4	4	3
Motivation	Q10	1 = not involved by the game 5 = always makes sure everyone is on point	5	5	5	4	3
	Q11 ONLY FOR PO	1 = poor/absent advices 5 = wise and helpful suggestions when is required	5				
	Q12	1 = doesn't express opinions during retrospective 5 = feels the retrospective fundamental to express opinions	5	5	5	5	5
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	5	5	5	5	5
	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
	Q15 ONLY FOR PO	Same evaluation as Q14 for the PO	5				

Figure 8: Screenshot della scheda da LibreOffice

3.1.4 Flusso di lavoro

Abbiamo deciso di utilizzare il [GitHub Workflow](#).

Nel nostro caso, abbiamo deciso che ai task delle US, se possibile, avremmo associato feature branch su cui sviluppare e completare i singoli task, per poi effettuare una merge request al branch main che sarebbe stata approvata in seguito.



Figure 9: Screenshot da Gitlab

3.2 Sprint 1

3.2.1 Sprint goal

Il nostro obiettivo fin da questo sprint è stato quello di portare un client che potesse effettuare ricerche di tweet per parola chiave.

Ci è sembrato un buon obiettivo da raggiungere nelle prime due settimane di lavoro.

3.2.2 Sprint backlog

- Come abitante di una bolla, voglio filtrare i tweet per parola chiave, così da vedere solo quello che interessa a me

Stima: 8 punti

- Come visitatore di pagine web vorrei poter visualizzare la homepage dell'applicazione

Stima: 13 punti

3.2.3 Definition of Done

Questo è stato uno dei punti dolenti di questo sprint: essendoci concentrati molto sul portare delle funzionalità al prof, abbiamo trascurato la DOD.

In generale, durante questo sprint abbiamo considerato una US *closed* quando era implementato ciò che ne era richiesto dalla US su taiga.

3.2.4 Burndown dello sprint



Figure 10: Screenshot da Taiga

3.2.5 Test

Purtroppo in questo sprint non siamo riusciti a scrivere nemmeno un test, abbiamo cercato di recuperare a partire dallo sprint 2.

3.2.6 Scelte progettuali

Questo è stato uno sprint importante anche per la scelta delle tecnologie da utilizzare per realizzare il prodotto.

Poichè alcuni di noi erano già familiari allo sviluppo di applicazioni web con React e Node, abbiamo deciso di realizzare una web application con le seguenti tecnologie:

- React per il frontend (con l'utilizzo di una libreria per la gestione dello stato, [Redux](#))
- [Tailwind CSS](#) come framework css
- Node per il backend (per interfacciarsi con le API di twitter)

Per fare in modo che tutti si trovassero allo stesso livello con le tecnologie scelte e con gli strumenti CAS, abbiamo creato degli *Storyless tasks* su Taiga, quali:

- Studio di react + redux
- Studio delle API di twitter (v2)
- Studio di Taiga

3.2.7 Sprint Review

Grazie alla Sprint Review siamo riusciti a capire che la nostra US riguardante la Homepage del sito fosse ridondante e non richiesta.

Ne abbiamo preso atto per gli sprint successivi.

3.2.8 Retrospettiva

In questo sprint abbiamo cominciato a capire la vera natura di Scrum oltre a conoscerci meglio tra di noi perché se non per Juan e Alex che si conoscevano da prima, tutti gli altri membri del team non si conoscevano tra loro.

Proprio perché non avevamo mai lavorato con Scrum, abbiamo commesso degli errori abbastanza gravi:

- Durante lo sprint planning abbiamo trascurato la DOD.
- Stessa cosa è successa con lo sprint goal.
- Poichè ci siamo concentrati tantissimo sullo scrivere codice, abbiamo trascurato del tutto la parte di testing, non abbiamo scritto alcun test.

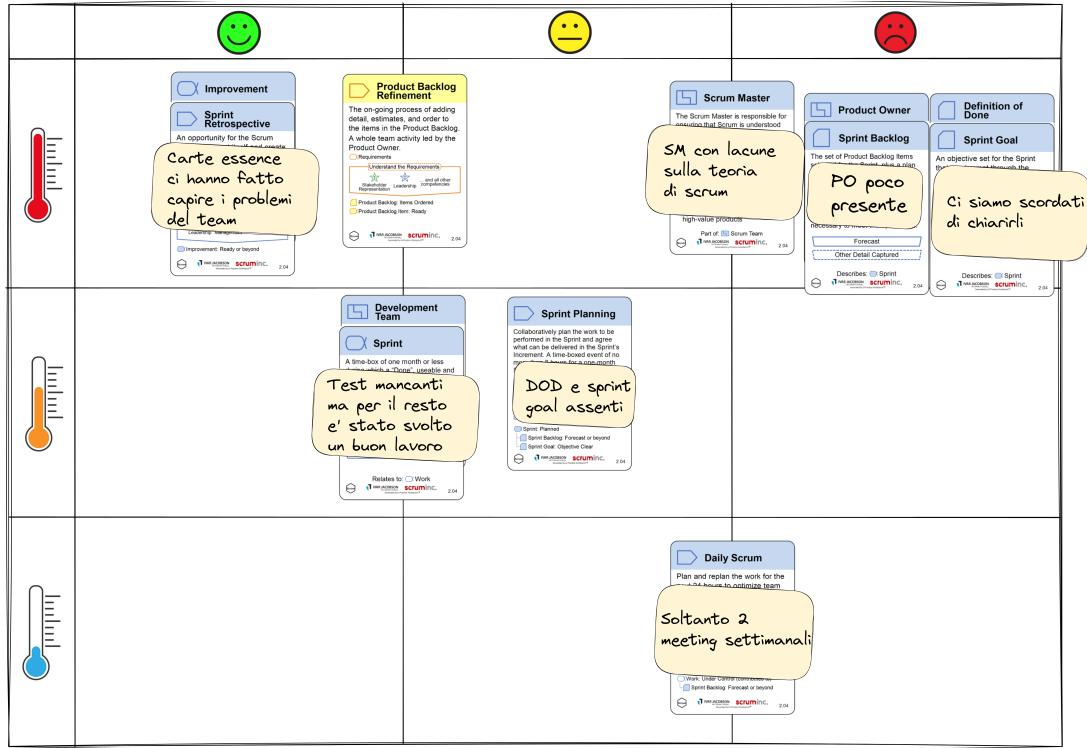


Figure 11: Immagine esportata da <https://excalidraw.com>

3.3 Sprint 2

3.3.1 Sprint goal

Abbiamo deciso di fissare il seguente obiettivo da raggiungere durante questo sprint:

Portare alla demo un client che permettesse di filtrare ricerche e visualizzare statistiche di sentiment analysis e di distribuzione temporale su tali ricerche.

È stato un obiettivo abbastanza impegnativo che per fortuna siamo riusciti a raggiungere, con qualche piccola mancanza.

3.3.2 Sprint backlog

- Come genitore apprensivo vorrei un modo per cercare i tweet di mio figlio per monitorarlo

Stima: 15 punti

- Come analista vorrei avere un grafico che mi permette di vedere il numero di tweet raccolti nel tempo, così da sapere gli andamenti di popolarità dei

topic

Stima: 13 punti

- Come genitore apprensivo vorrei un modo per cercare i tweet di mio figlio e visualizzarli sulla mappa

Stima: 13 punti

- Come analista di dati vorrei un grafico a torta sul sentiment dei tweet, per farmi un'idea di cosa ne pensino gli utenti

Stima: 28 punti

- Come storico, voglio poter scegliere un intervallo di date, per selezionare i tweet del solo periodo che sto studiando

Stima: 14 punti

- Come binge tweeter, voglio poter visualizzare piu' di 10 tweet alla volta

Stima: 15 punti

3.3.3 Definition of Done

Durante questo sprint abbiamo considerato una US *closed* quando venivano rispettati i seguenti punti:

- Il codice che la implementa è stato sufficientemente commentato.
- Il codice rispecchia ciò che è scritto nella us.
- Il codice è stato visionato da almeno un altro sviluppatore (attraverso le [merge requests](#)).

3.3.4 Burndown dello sprint

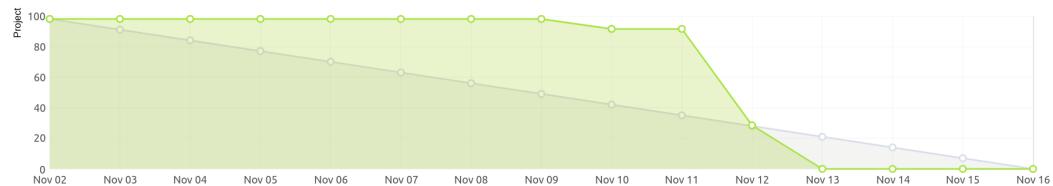


Figure 12: Screenshot da Taiga

Analizzando il burndown, si riescono a notare:

- Poca praticità con l'utilizzo di Taiga.

- Avendo finito prima del previsto le US di Taiga, abbiamo deciso di non prenderne altre dal product backlog perchè sapevamo che non saremmo riusciti a completarle.

Piuttosto ci siamo concentrati sul consolidare le nostre conoscenze di CAS e del testing per JS ([CI/CD Pipelines](#), [Sonarqube](#), [React testing library](#) e [Mock service worker](#)) in vista del prossimo sprint.

3.3.5 Test

A seguire riportiamo uno tra i vari test che abbiamo incluso:
given a query should respond with up to max_results elements:

Questo test assicura che quando è stato settato il parametro di query *max_results* vengono restituiti, dalla nostra API, al più *max_results* tweet.

3.3.6 Scelte progettuali

Durante questo sprint abbiamo deciso di utilizzare:

- [Leaflet](#) per la visualizzazione della mappa, inizialmente il nostro team di sviluppatori aveva scelto Google Maps, ma per mantenere coerenza con le altre tecnologie *open source* abbiamo fatto un *refactor* per passare a [Leaflet](#).
- [Multilang-sentiment](#) come libreria per la sentiment analysis.

3.3.7 Sprint Review

Grazie alla sprint review siamo riusciti a capire alcune lacune del nostro prodotto, tra cui:

- ci è stato fatto notare il fatto che la nostra visualizzazione fosse “limitata”, il professore voleva una visualizzazione di un grafico a barre dove nell’asse x ci fossero solo i giorni, al più avrebbe voluto un grafico a barre per minuti.
- ci sono stati richiesti i sentimenti dei singoli tweet (per avere una visione più chiara delle statistiche, e anche perchè così facendo si riuscirebbe a diminuire l’impressione di [black box](#))
- la nostra ricerca di tweet era limitata a soli 99 tweet, quando le API di twitter ci permettono di chiedere 100 tweet alla volta, per cui ci è stato chiesto di aggiungere la possibilità di scegliere 100 tweet.

Abbiamo preso atto delle note del professore e le abbiamo implementate nello sprint a seguire.

3.3.8 Retrospettiva

Questo sprint è andato molto bene rispetto a quello precedente, siamo riusciti a superare le criticità che erano sorte a fine della restrospettiva dello sprint 1.

Le cose su cui avremmo potuto fare di meglio sono:

- Lo sprint planning: e' andato abbastanza bene se non fosse per il fatto che abbiamo dovuto tenere un secondo incontro per lo sprint planning poichè non eravamo riusciti, per motivi di tempo, a creare i task necessari per le US.
- La DOD: Abbiamo iniziato a fare test (del backend) ma non li abbiamo inclusi nella DOD perché non sapevamo quanto sarebbe stato faticoso interfacciarsi con le librerie/framework di testing.

Il nostro PO è stato un po' assente a causa lavoro ma in fin dei conti non ne ha risentito lo sprint backlog.

Il nostro SM è stato molto presente e ha comunicato molto con il team.

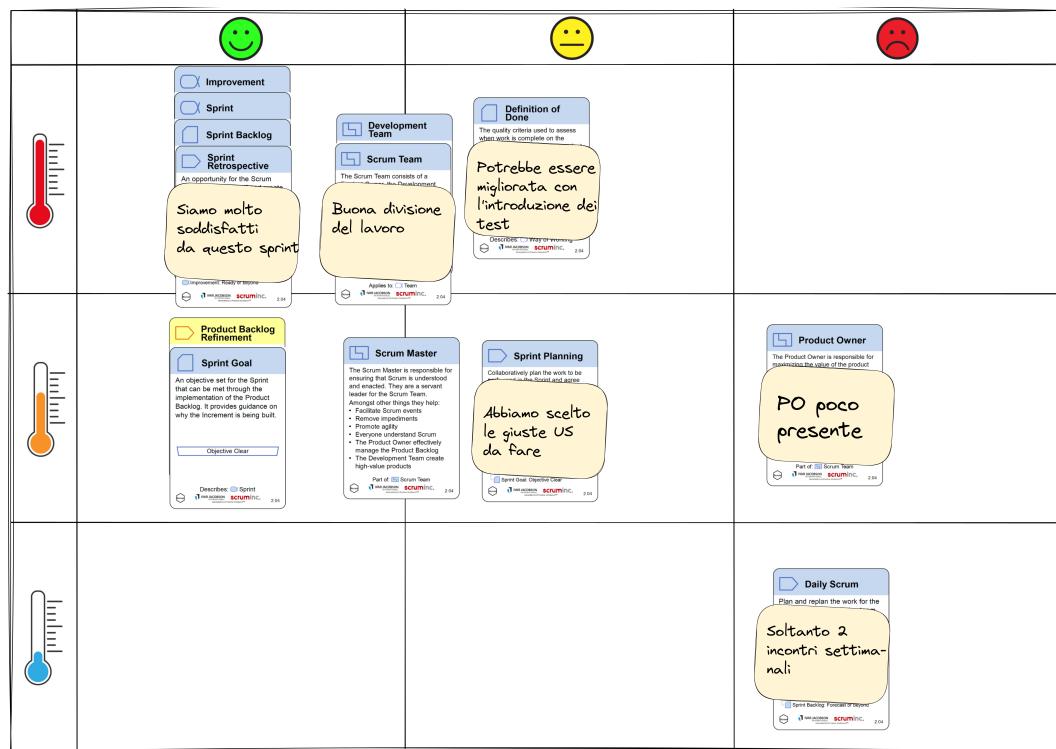


Figure 13: Immagine esportata da <https://excalidraw.com>

3.4 Sprint 3

3.4.1 Sprint goal

Lo sprint goal di questo sprint è stato il seguente:

Avere un client che permettesse di visualizzare statistiche utili, interessanti sulle ricerche effettuate. Inoltre il client deve supportare la visualizzazione dei campioni dell'eredità.

A questo punto, per fortuna, eravamo messi abbastanza bene dal punto di vista delle funzionalità implementate, questo ci ha permesso di fissare uno sprint goal "onesto".

3.4.2 Sprint backlog

- Come membro dell'accademia della crusca vorrei vedere le parole dei tweet selezionati su una term cloud, così da poter vedere le parole più utilizzate
Stima: 15 punti
- 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
Stima: 32 punti

A seguire sono riportate le US che non abbiamo completato

- Come giocatore di scacchi voglio sfidare gruppi di persone in rete, per giocare partite le cui mosse verranno scelte a maggioranza
Stima: 18 punti
- Come giocatore di scacchi, voglio raccogliere i tweet che rispondono alla mia mossa, per scegliere e visualizzare la mossa scelta dalla maggioranza
Stima: 23 punti

A seguire sono riportati gli issue che abbiamo risolto, sorti durante la sprint review dello sprint 2.

- La card dei singoli tweet va estesa per includere la sentiment analysis del tweet
- Gli endpoint che raccolgono i tweet devono fare in modo di poter ritornare fino a 100 tweets
- Il diagramma a barre va esteso con la possibilità di avere i minuti come unità di tempo
- Il diagramma a barre va esteso con la possibilità di avere i giorni come unità di tempo

3.4.3 Definition of Done

La DOD è rimasta uguale a quella dello sprint scorso, abbiamo provato ad aggiungere anche il seguente punto:

- Il codice che implementa la US è coperto da test

Tuttavia per motivi di tempo (il setup per eseguire i test nel frontend non è banale), non siamo riusciti a testare molti componenti del frontend, ci siamo limitati a testarne un paio.

3.4.4 Burndown dello sprint

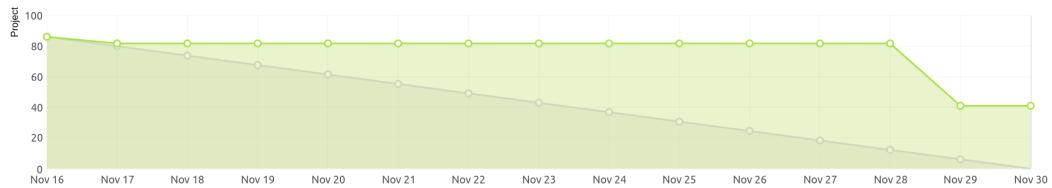


Figure 14: Screenshot da Taiga

Analizzando il burndown, si riescono a notare i seguenti punti:

- Sovvraccarico di lavoro.
- Non siamo riusciti a completare tutte le US e questo ci ha fatto accumulare debito tecnico che si è concretizzato nell'implementazione di pochi test per la nostra codebase.

3.4.5 Test

A seguire riportiamo due esempi di test del backend e del frontend, rispettivamente :

should return an array of wordcloud info if input tweets are provided:

Questo test assicura che venga creato un array contenente le informazioni di wordcloud che ci interessano.

SearchForm component should show filters when they're enabled:

Questo test assicura che il componente SearchForm renda visibili i filtri di ricerca se vengono abilitati

3.4.6 Scelte progettuali

Durante questo sprint abbiamo deciso di utilizzare:

- Abbiamo deciso di reperire le informazioni sulla ghigliottina dall'account twitter che posta tutti i giorni le risposte giuste e i relativi campioni: <https://twitter.com/quizzettone>

- Al backend è stato eseguito un refactor per fare in modo che venisse utilizzato [node-twitter-api-v2](#) come client per interfacciarsi con le API di twitter, questo ci ha aiutato a gestire al meglio gli stream.
 - [react-chessboard](#) come componente per la scacchiera lato frontend e [chess.js](#) come libreria per gestire la logica degli scacchi.
 - [chess-image-generator](#) come libreria per la generazione delle immagini delle scacchiere dati FEN, PGN o array (a noi è venuto comodo utilizzare `.loadFEN(fen)`).
 - Poiché non avevamo a disposizione nessuna macchina su cui fare il deployment della nostra applicazione, abbiamo deciso richiedere uno spazio web al DISI ([da qui](#)).
- Ci è stato assegnato l'indirizzo <http://site212236.tw.cs.unibo.it>

3.4.7 Pipeline

In questo sprint, Juan si è occupato di implementare la pipeline per la continuous integration, testing e deployment. (a tal proposito è stato molto utile consultare la documentazione ufficiale di gitlab: [pipelines](#))

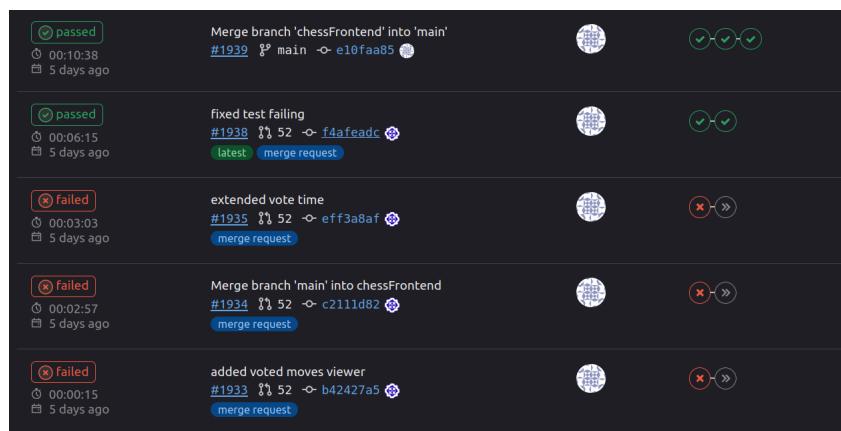


Figure 15: Screenshot da gitlab

3.4.8 Sprint Review

Il 5/12/2022 abbiamo tenuto il secondo incontro con il professore per lo stato di avanzamento del prodotto durante lo sprint 3.

Durante la discussione il professore è sembrato entusiasta e soddisfatto per il nostro stato di avanzamento.

La discussione col professore è andata molto bene e la demo è piaciuta.

3.4.9 Retrospectiva

Questo sprint è andato bene anche se non siamo riusciti a completare tutte le US dello sprint backlog perché abbiamo sottovalutato la complessità delle US degli scacchi.

Le cose da migliorare sono state:

- La stima dell'effort necessario per completare le US
- La situazione dei test del frontend (anche se e' stato veramente complesso settare l'ambiente di test e questo ci ha rallentati parecchio) e del backend

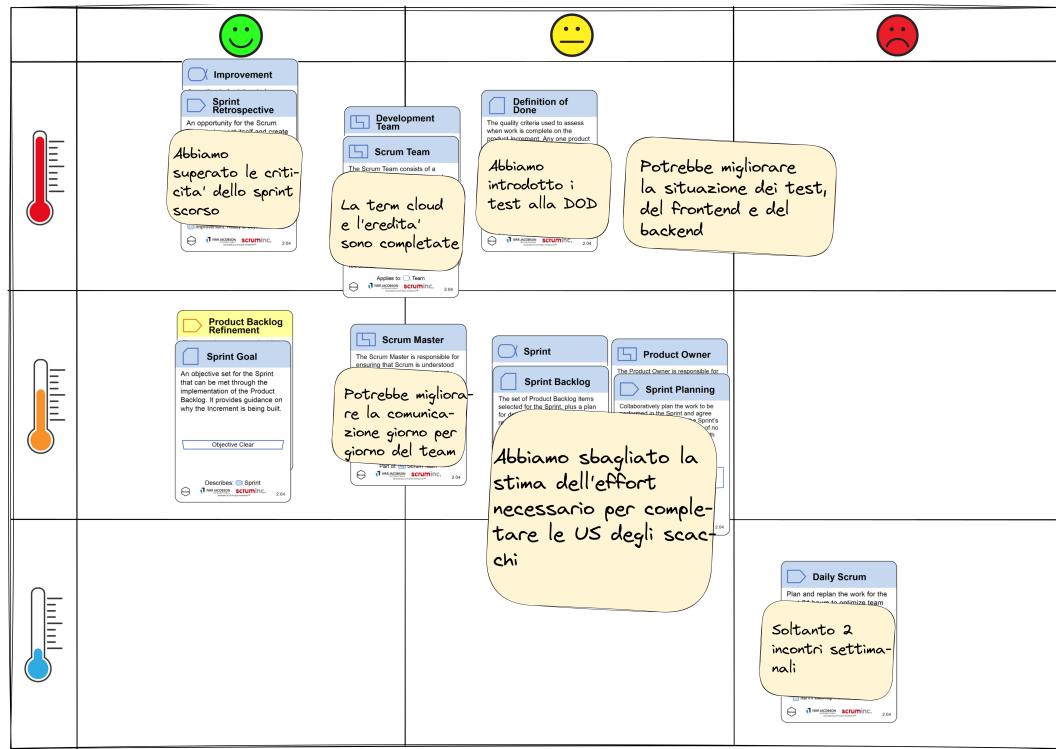


Figure 16: Immagine esportata da <https://excalidraw.com>

3.5 Sprint 4

3.5.1 Sprint goal

Le nostre intenzioni sono state fin dall'inizio quelle di consegnare il progetto a Dicembre, perciò questo sprint ha avuto come obiettivo:

Avere un client che permetta di giocare a scacchi tramite l'utilizzo di tweet e che permetta di ottenere informazioni interessanti sul fantacitorio.

3.5.2 Sprint backlog

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

Stima: 18 punti

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

Stima: 23 punti

- Come spettatore del fantacitorio voglio poter visualizzare la classifica emendabile settimanale / di sempre dei politici.

Stima: 26 punti

- Come spettatore del fantacitorio voglio poter vedere le squadre dei miei amici

Stima: 19 punti

- Come spettatore del fantacitorio voglio poter sfogliare le immagini delle squadre registrate

Stima: 24 punti

3.5.3 Definition of Done

La DOD è rimasta uguale a quella dello sprint scorso. Siamo riusciti a testare gran parte del backend, purtroppo non possiamo dire lo stesso del frontend: per testare tutti i componenti del frontend, ci sarebbe voluto un altro sprint.

3.5.4 Burndown dello sprint



Figure 17: Screenshot da Taiga

3.5.5 Test

A seguire riportiamo due esempi di test del backend e del frontend, rispettivamente :

should return an array of objects with the politicians and their summed points with full names:

Questo test assicura che venga creato un array contenente le informazioni per la classifica del fantacitorio che ci interessano.

form utils' is ValidDateRange function should work as expected w/valid ranges:

Questo test assicura che il la funzione isValidDateRange (utilizzata quando viene effettuata una ricerca, per controllare che l'intervallo delle date sia valido) validi i range di date correttamente.

3.5.6 Scelte progettuali

- Abbiamo deciso di reperire le informazioni sul fantacitorio dall'account twitter del fantacitorio: https://twitter.com/Fanta_citorio
- Per fare in modo di supportare la richiesta del prof della classifica non solo dell'ultima settimana, abbiamo utilizzato <https://developer.twitter.com/en/docs/twitter-api/users/lookup/api-reference/get-users-id>.
Questo endpoint ci permette di ricavare i tweet di un utente senza limitazioni sull'arco temporale (i.e. siamo riusciti a soddisfare la richiesta del prof senza utilizzare basi di dati di appoggio).

3.5.7 Retrospettiva

Questo sprint è andato bene, siamo riusciti a completare lo sprint goal che ci eravamo prefissati (i.e. consegnare il progetto prima di natale), ma ci sono state comunque delle cose che avremmo potuto fare meglio:

- La comunicazione del team: abbastanza cruciale per questo ultimo sprint, è stata scarsa da parte di alcuni componenti del team.
- La DOD: Abbiamo testato al dettaglio il backend, purtroppo non siamo riusciti a fare lo stesso per il frontend poiché ci sarebbe voluto un altro sprint per aumentare la coverage dei nostri componenti.

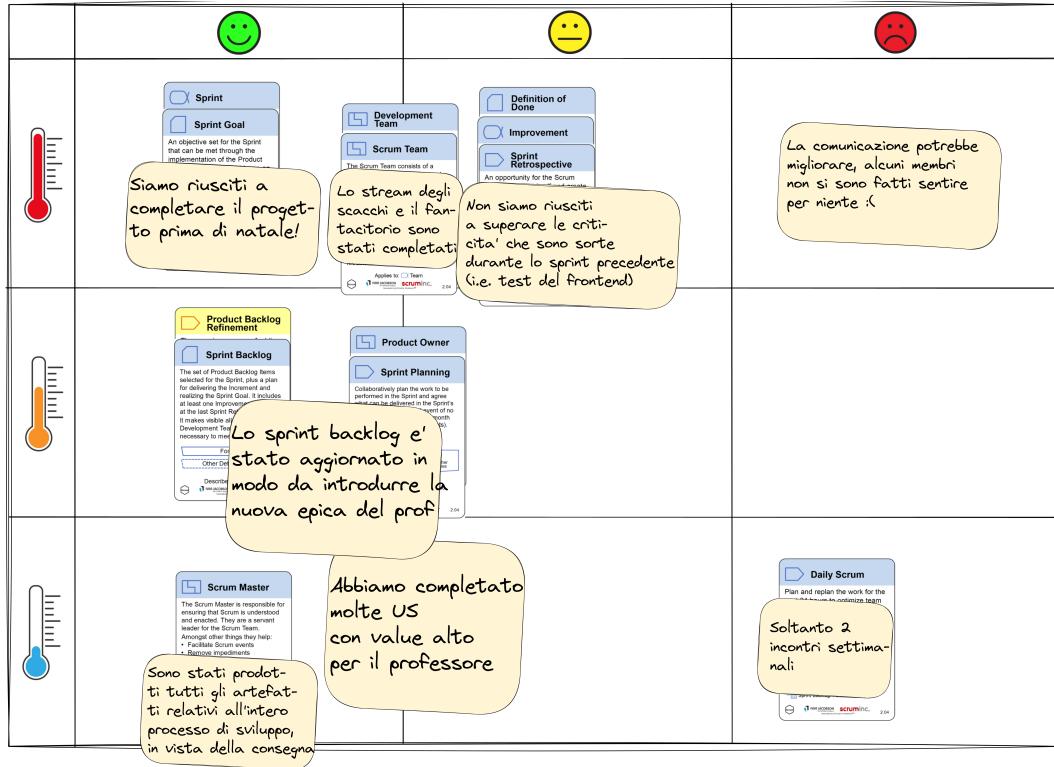


Figure 18: Immagine esportata da <https://excalidraw.com>

3.6 Alcuni dati sul processo

3.6.1 Sonarqube

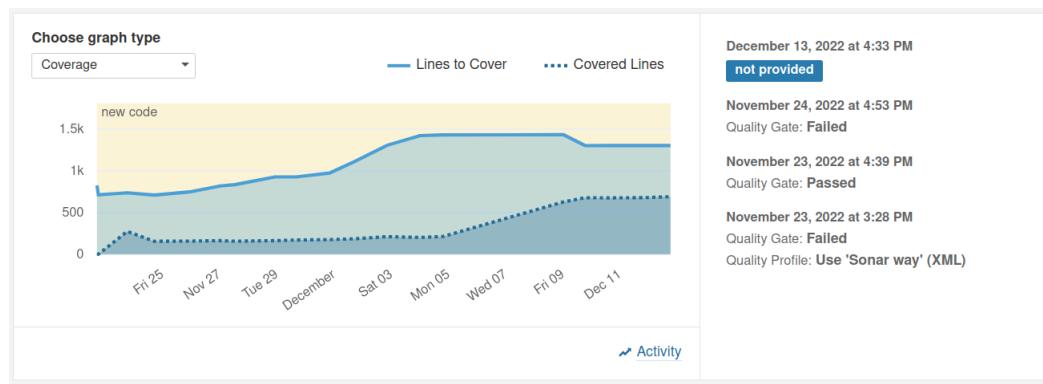
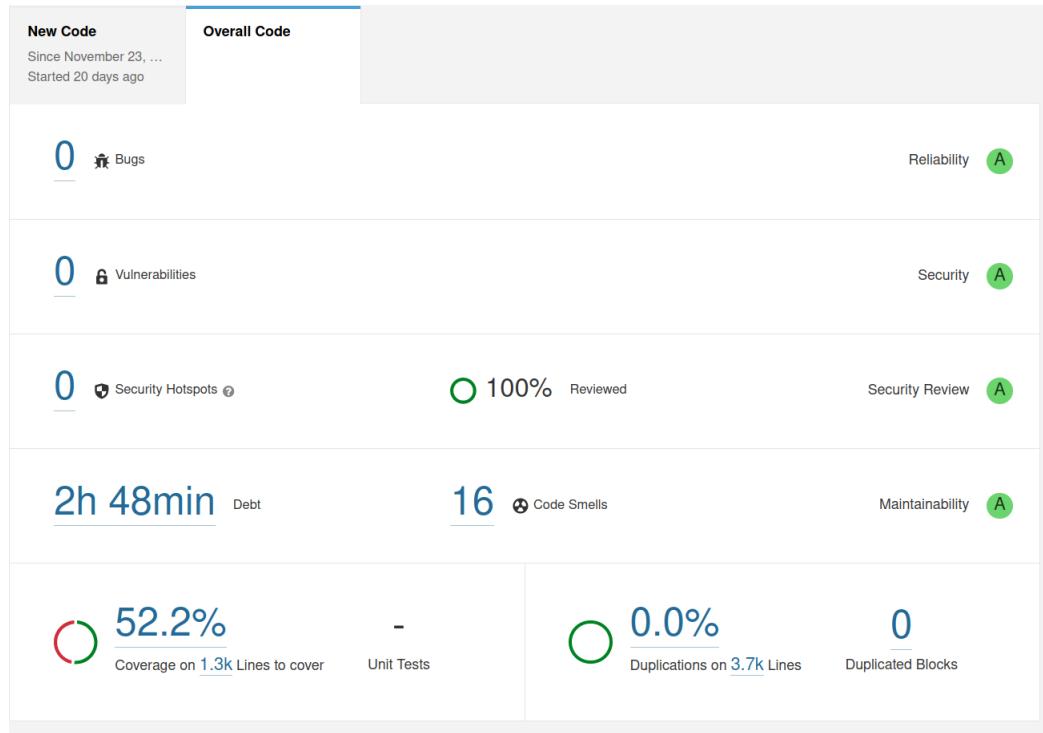


Figure 19: Screenshot della dashboard dell'istanza di sonarqube su qube.hjkl.gq

3.6.2 Gitinspector

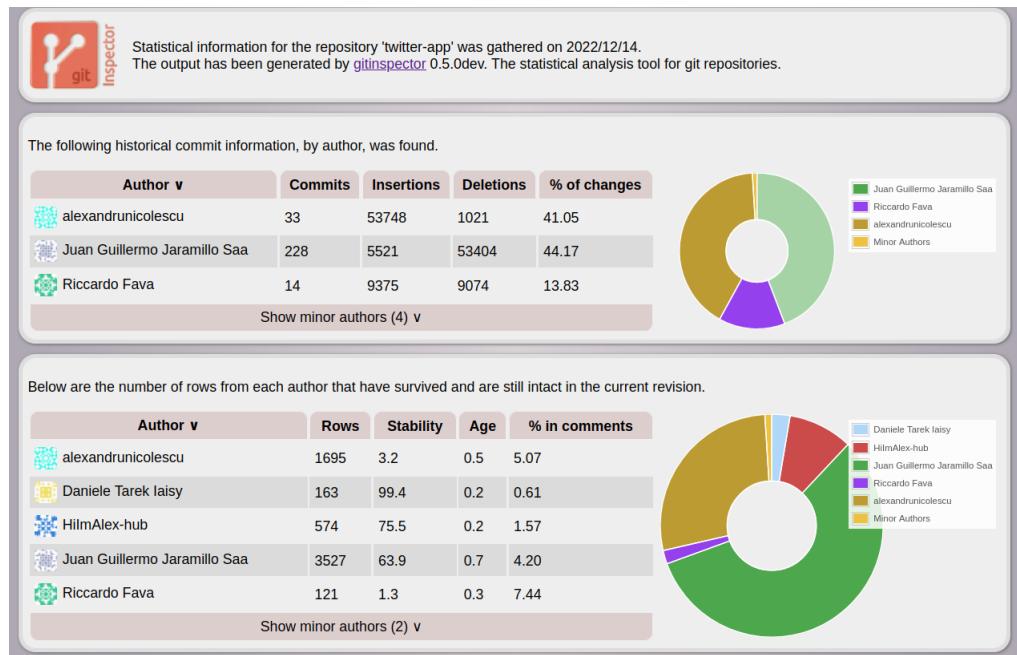


Figure 20: Screenshot dell'output di gitinspector

3.6.3 Burndown finale

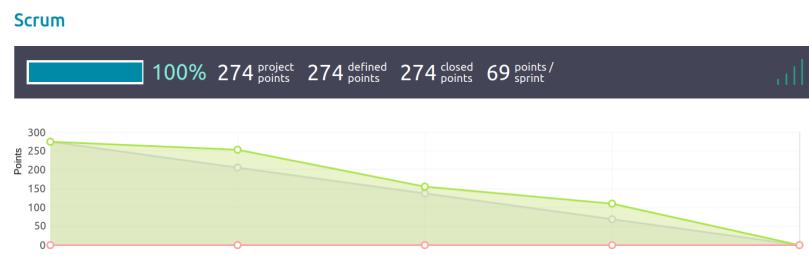


Figure 21: Screenshot da Taiga

4 Artefatti

- **Link del sito di produzione:** <http://site212236.tw.cs.unibo.it/>
- **Link all'account twitter dell'app:** <https://twitter.com/ingsw2022team3>
- **Video della demo:** <https://peertube.devol.it/w/7AnJMxZ3cREsUiyzgREu5X>
- **Taiga del progetto:** <https://taiga.hjkl.gq/project/ingsw2022-team3/timeline>
- **Wiki del progetto:** <https://taiga.hjkl.gq/project/ingsw2022-team3/wiki/home>
- **Retrospective:** <https://taiga.hjkl.gq/project/ingsw2022-team3/wiki/retrospective>
- **Gitlab:** <https://git.hjkl.gq/team3/twitter-app>
- **Continuous Integration - Continuous Testing - Continuous Deployment:** <https://git.hjkl.gq/team3/twitter-app/-/blob/main/.gitlab-ci.yml>
- **Eseguire in locale:** <https://git.hjkl.gq/team3/twitter-app/-/blob/main/README.md>
- **Sonarqube:** link a sonarqube