

Tweet Analysis

Relazione finale

Cheikh Ibrahim · Zaid

PO Operativo

Matricola: 0000974909

Xia · Tian Cheng

Scrum master

Matricola: 0000975129

Lee · Qun Hao Henry

Developer

Matricola: 0000990259

Paris · Manuel

Developer

Matricola: 0000997526

Anno accademico

2022 — 2023

Corso di Ingegneria del Software

Alma Mater Studiorum · Università di Bologna

Indice

1	Descrizione del prodotto	2
1.1	Scope	2
1.2	Casi d'uso	2
1.3	Diagramma delle classi	2
2	Descrizione degli sprint	3
2.1	Sprint 1	3
2.1.1	Sprint goal	3
2.1.2	Backlog	3
2.1.3	Esito sprint	5
2.1.4	Sprint review	5
2.1.5	Retrospettiva	6
2.2	Sprint 2	8
2.2.1	Sprint goal	8
2.2.2	Backlog	8
2.2.3	Esito sprint	10
2.2.4	Sprint review	10
2.2.5	Retrospettiva	11
2.3	Sprint 3	12
2.3.1	Sprint goal	12
2.3.2	Backlog	12
2.3.3	Esito sprint	14
2.3.4	Sprint review	14
2.3.5	Retrospettiva	15
2.4	Sprint 4	16
2.4.1	Sprint goal	16
2.4.2	Backlog	16
2.4.3	Esito sprint	18
2.4.4	Retrospettiva	19
3	Descrizione del processo	20
3.1	Team building	20
3.1.1	Scrumble	20
3.1.2	Escape the Boom	20
3.2	Gitinspector	20
3.3	Retrospettiva finale	20
3.4	Deployment	20
4	Artefatti	21

1 Descrizione del prodotto

1.1 Scope

1.2 Casi d'uso

1.3 Diagramma delle classi

2 Descrizione degli sprint

Sono stati svolti quattro sprint della durata di 14 giorni ciascuno.

La stima dei punti delle user stories è stata effettuata con una scala da 0 a 10, valutando separatamente il frontend dal backend. Il punteggio complessivo è quindi ottenuto dalla somma di quest'ultimi.

2.1 Sprint 1

2.1.1 Sprint goal

Lo sprint è stato principalmente dedicato a studiare le API di Twitter e produrre le prime funzionalità per la visualizzazione e l'analisi dei tweet.

In particolare le feature pianificate per lo sprint sono state:

- Ricerca di tweet per username
- Ricerca di tweet per hashtag
- Analisi dei tweet tramite componenti grafiche:
 - Grafico a torta per il sentiment analysis
 - Grafico a barre per la frequenza dei Tweet
 - Word cloud con le parole più usate

2.1.2 Backlog

US: Come utente interessato ai tweet, voglio poter cercare dei tweet per hashtag per leggerli.	Punti: 8 (3 frontend + 5 backend)
---	---

DOD: L'utente, cercando un hashtag in un apposito textbox, è in grado di leggere tutti i tweet correlati visualizzando: nome account Twitter, username, immagine profilo, contenuto tweet (testo + foto e video), data e ora, luogo (se applicabile), numero like, numero commenti, numero retweet.

Test: Richiamare l'API implementata, verificare che il formato sia corretto e che il contenuti dei tweet contenga l'hashtag ricercato.

US: Come utente interessato ai tweet, voglio poter cercare dei tweet per nome utente per leggerli.	Punti: 8 (3 frontend + 5 backend)
---	---

DOD: L'utente, cercando un nome utente in un apposito textbox, è in grado di leggere tutti i tweet correlati visualizzando: nome account Twitter, username, immagine profilo, contenuto tweet (testo + foto e video), data e ora, luogo (se applicabile), numero like, numero commenti, numero retweet.

Test: Richiamare l'API implementata, verificare che il formato sia corretto e che l'autore dei tweet sia quello ricercato.

US: Come analista,
voglio poter analizzare il sentimento
per stabilire se un tweet è positivo o meno. **Punti:** 9
(2 frontend + 7 backend)

DOD: L'utente, dato un tweet, vede se è positivo, negativo o neutro tramite immagine o testo.

Test: Analizzare frasi di cui è noto il sentimento.

US: Come analista,
voglio vedere un grafico a barre
per vedere il numero di tweet nell'unità di tempo. **Punti:** 2
(2 frontend)

DOD: L'utente apre una pagina web contenente il grafico a barre con il numero di tweet nell'unità di tempo.

Test: Manualmente verificare che il grafico sia corretto.

US: Come analista,
voglio vedere un grafico a torta
per vedere il rapporto di sentiment positivi, negativi e
neutri. **Punti:** 2
(2 frontend)

DOD: L'utente apre una pagina web contenente il grafico a torta con sentiment positivi, negativi e neutri.

Test: Manualmente verificare che il grafico sia corretto.

US: Come analista,
voglio vedere una term cloud
per vedere le parole più utilizzate nei tweet. **Punti:** 4
(2 frontend + 2 backend)

DOD: L'utente apre una pagina web contenente una term cloud con le parole più utilizzate.

Test: Manualmente verificare che il grafico sia corretto.

2.1.3 Esito sprint

Lo sprint è terminato con la conclusione di tutte le user stories pianificate.

Il lavoro è risultato omogeneo, con un piccolo scostamento rispetto all'andamento ideale che però non è stato causa di imprevisti o ritardi.

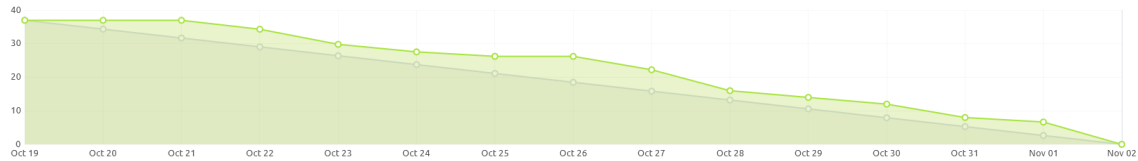


Figura 1: Burndown sprint 1

Anche le ore di lavoro (con il tempo stimato in quanto allo sprint 1 non è stato effettuato il monitoraggio) sono risultate in linea con il monte ore previsto.

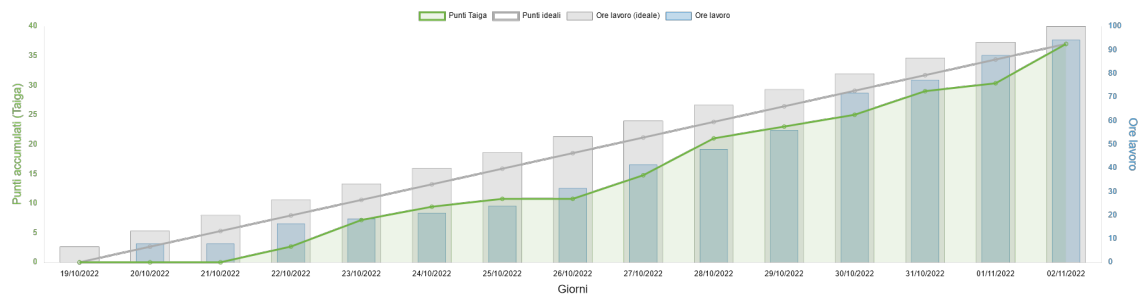


Figura 2: Progresso dei punti (asse a sinistra) e ore di lavoro (asse a destra)

2.1.4 Sprint review

Alla sprint review è stato chiesto dal cliente l'implementazione delle seguenti feature:

- Ricerca di tweet per intervallo temporale
- Possibilità di selezionare il numero di tweet da estrarre con una singola richiesta

2.1.5 Retrospettiva

Pre-retrospettiva

Alla pre-retrospettiva effettuata a metà sprint, sono emerse le seguenti problematiche:

- Tempo dedicato al lavoro non sufficiente
- Mancanza di comunicazione nel team
- Task e user stories hanno descrizioni poco dettagliate e facilmente fraintendibili

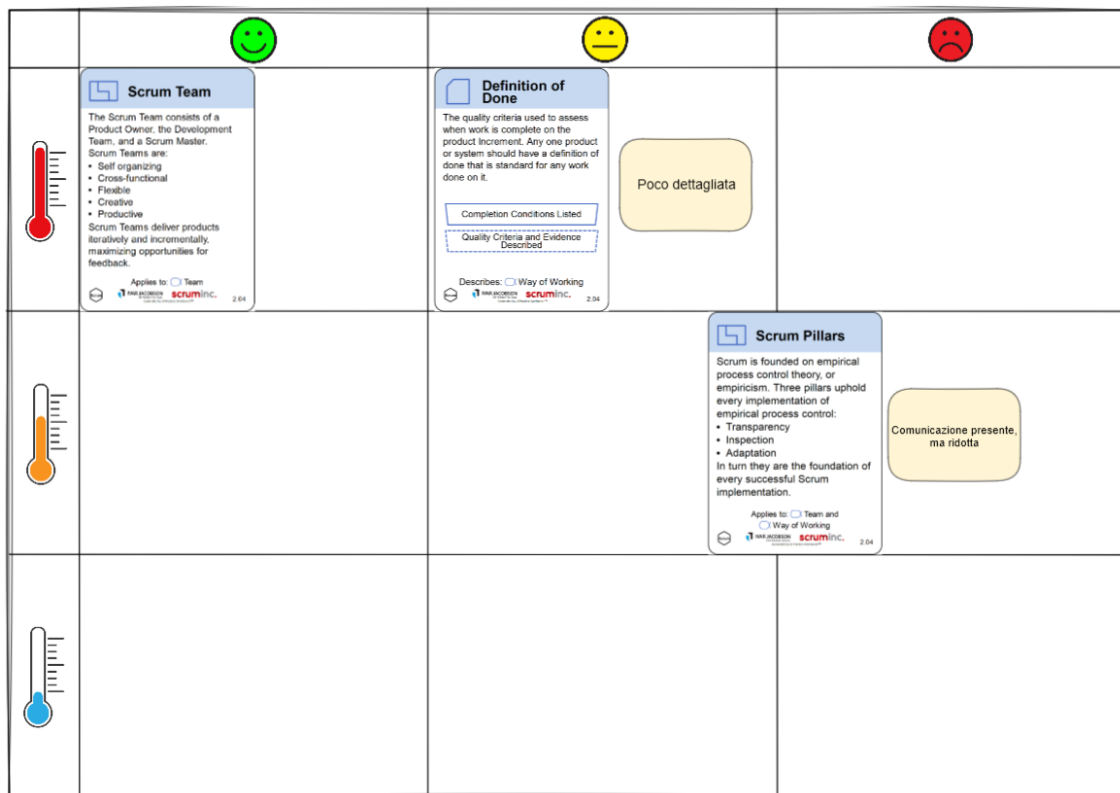


Figura 3: Pre-retrospettiva del 27/10/2022

Retrospettiva

Alla retrospettiva di fine sprint, gli aspetti positivi evidenziati dal team sono:

- Il deliverable presentato alla sprint review è risultato adeguato e funzionante
- Il team e i ruoli di ciascuno sono ben definiti

Sono invece stati portati all'attenzione del team i seguenti problemi:

- Non è stato effettuato un monitoraggio delle ore di lavoro (che è stato quindi stimato sulla base sui daily scrum virtuali)

- I problemi evidenziali alla pre-retrospettiva non sono ancora stati risolti

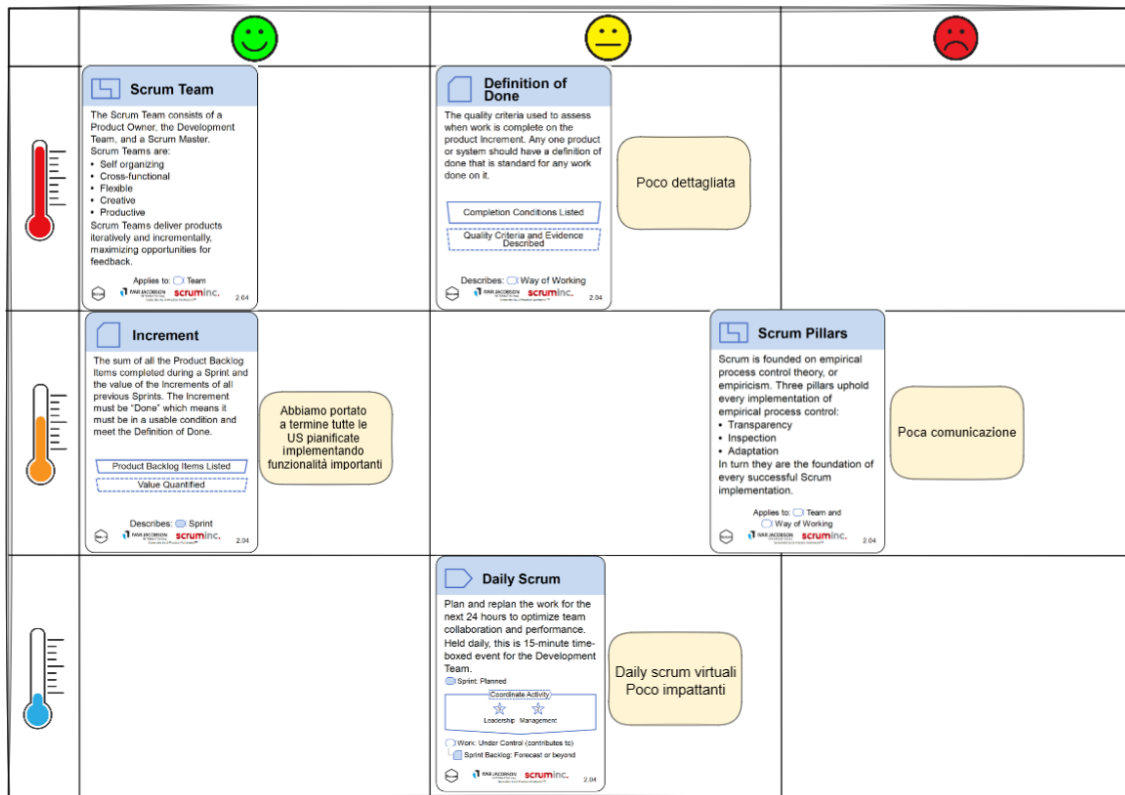


Figura 4: Retrospettiva del 01/11/2022

2.2 Sprint 2

2.2.1 Sprint goal

L'obiettivo dello sprint è stato quello di concludere l'epica riguardante la visualizzazione e l'analisi dei tweet.

Nello specifico, sono state implementate le seguenti funzionalità:

- Ricerca di tweet per intervallo temporale (richiesto dal cliente allo sprint review precedente)
- Ricerca di un determinato numero di tweet con una singola ricerca (richiesto dal cliente allo sprint review precedente)
- Ricerca di tweet per parola chiave
- Mappa per visualizzare la posizione dei tweet con geolocalizzazione
- Raccolta di tweet in tempo reale

2.2.2 Backlog

US: Come utente interessato a vedere tweet, voglio scegliere un numero di tweet da poter caricare in una volta per poterli analizzare in modo aggregato. **Punti: 5**
(3 frontend + 2 backend)

DOD: Possibilità di ricercare un largo numero di tweet scrivendone la quantità in un textbox.

Tale quantità serve per la ricerca iniziale e anche per mostrare pagine successive. È possibile inoltre ricercare un numero di tweet inizialmente per poi cambiare quantità e cercare una pagina successiva (esempio: ricerca di 150 tweet iniziali, poi viene modificata la quantità (dallo stesso textbox) in 20 e si preme su “Pagina successiva”. Il numero di tweet così mostrato diventa 170).

Test: Verificare che il numero di tweet raccolto corrisponda con quello richiesto.

US: Come utente interessato a vedere tweet, voglio scegliere un intervallo di tempo in cui raccogliere tweet per analizzarne le tendenze storiche. **Punti: 5**
(3 frontend + 2 backend)

DOD: Possibilità di ricercare dei tweet dato un intervallo temporale. Non deve essere possibile cercare tweet nel futuro.

Non deve essere possibile cliccare su “Prossima pagina” quando non ci sono più tweet da visualizzare.

Test: Verificare che i tweet raccolti siano compresi nell'intervallo temporale.

US: Come utente interessato a vedere tweet,
voglio poter cercare dei tweet per parola chiave
per vedere cosa ne pensa la gente a riguardo.

Punti: 4
(2 frontend + 2 backend)

DOD: Possibilità di cercare tweet per parola o frase chiave. I grafici già presenti devono funzionare anche con questa ricerca.

Test: Richiamare l'API implementata, verificare che il formato sia corretto e che il contenuti dei tweet contenga la parola chiave ricercata.

US: Come utente interessato ai tweet,
voglio poter visualizzare su una mappa la posizione dei
tweet cercati
per avere un'idea della località dalla quale sono stati
pubblicati.

Punti: 7
(3 frontend + 4 backend)

DOD: Possibilità di visualizzare una mappa con le posizioni dei tweet ricercati.
Se sono presenti più tweet nella stessa zona è possibile aggregarli (in base alla distanza) e mostrare un unico valore, ovvero il numero di tweet in tale zona. La mappa deve essere sempre visibile anche durante lo scorrimento della pagina (come i grafici).

Test: Manualmente verificare che la mappa contenga i marker quando sono presenti tweet con geolocalizzazione.

US: Come utente,
voglio vedere la posizione di tutti i tweet di una data
persona
per conoscere i suoi spostamenti.

Punti: 5
(5 frontend + 0 backend)

DOD: Possibilità di inserire il nome utente di una persona e visualizzare su una mappa le posizioni dei suoi tweet e i suoi spostamenti.
Per gli spostamenti si mostrano sulla mappa delle frecce basandosi sulla posizione e sulla data del tweet.

Test: Manualmente verificare che la mappa contenga i marker quando sono presenti tweet con geolocalizzazione.

US: Come lettore di tweet,
voglio poter vedere i tweet che ricerco in tempo reale
per sapere cosa la gente posta.

Punti: 10
(3 frontend + 7 backend)

DOD: Poter scrivere una ricerca e, al click di un pulsante "Live", vedere tutti i tweet pubblicati in tempo reale a partire da quel momento.

Test: Verificare che il socket implementato restituisca i tweet ricercati, quando disponibili.

2.2.3 Esito sprint

Lo sprint è terminato con la conclusione di tutte le user stories pianificate.

Il lavoro si è svolto in linea con l'andamento ideale dei punti. Si è osservato una leggera sovrastima delle user stories che ha portato alla conclusione anticipata (di un giorno) dello sprint.

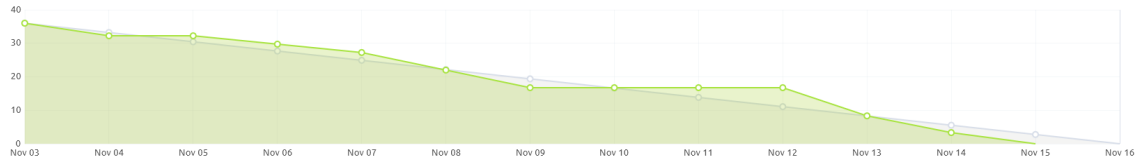


Figura 5: Burndown sprint 2

Le ore di lavoro rispettano il monte ore e sono state distribuite principalmente in due periodi di maggiore produttività.

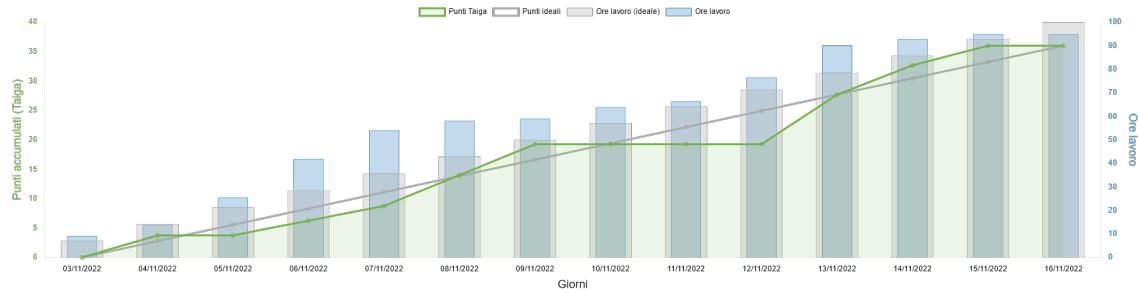


Figura 6: Progresso dei punti (asse a sinistra) e ore di lavoro (asse a destra)

2.2.4 Sprint review

Alla sprint review non sono emerse nuove richieste da parte del cliente.

2.2.5 Retrospettiva

Pre-retrospettiva

Alla pre-retrospettiva effettuata a metà sprint, sono emerse le seguenti problematiche:

- Mancanza di test adeguati per il frontend
- Le user stories sono state sovrastimate

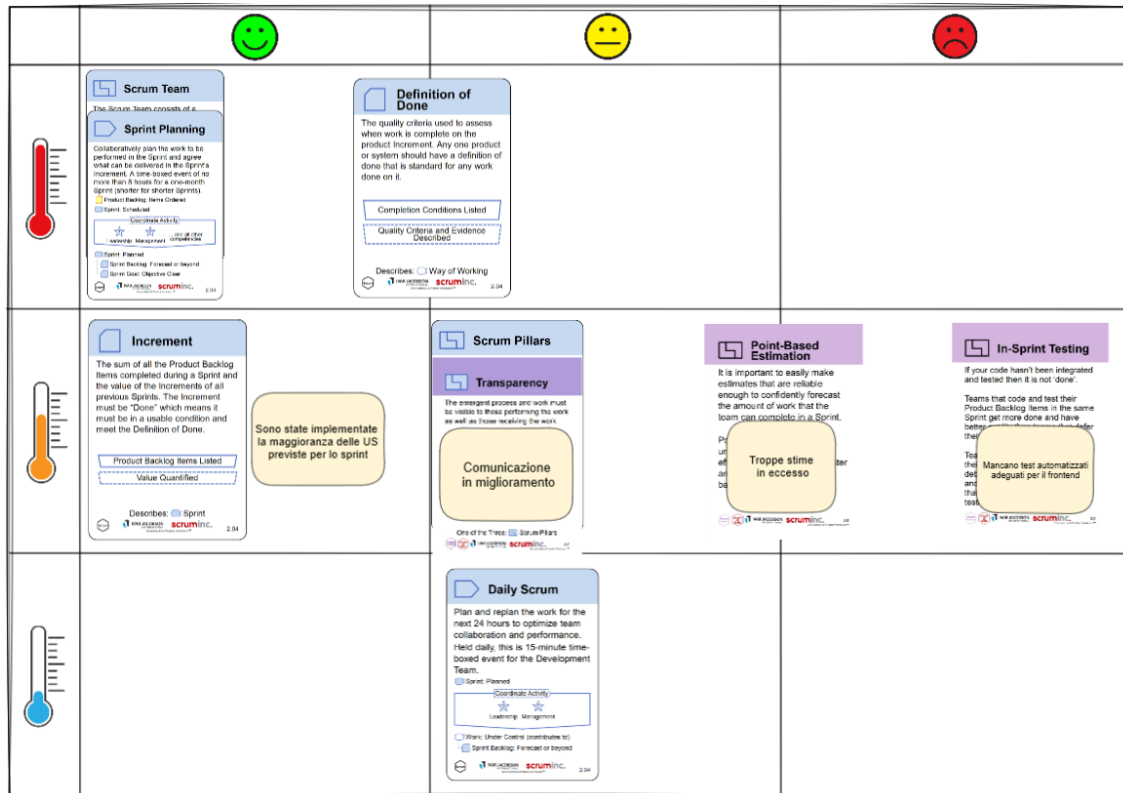


Figura 7: Pre-retrospettiva del 10/11/2022

Retrospettiva

Alla retrospettiva di fine sprint sono state confermate le problematiche della pre-retrospettiva.

2.3 Sprint 3

2.3.1 Sprint goal

Per lo sprint è stata pianificata l'implementazione delle epiche riguardanti i giochi televisivi (L'Eredità e Reazione a Catena) e l'inizio dell'epica riguardante lo scacchi.

In particolare sono state previste le seguenti funzionalità:

- Visualizzare gli utenti che tentano a indovinare la parola del giorno
- Visualizzare la parola vincente
- Visualizzare gli utenti che indovinano la parola
- Visualizzare la posizione degli utenti che tentano di indovinare
- Possibilità di muovere le proprie pedine su una scacchiera

2.3.2 Backlog

US: Come spettatore de #leredita, voglio raccogliere i tweet di chi prova a indovinare la ghigliottina, per visualizzare, in ordine temporale, tutti coloro che provano ad indovinare.

Punti: 5
(2 frontend + 3 backend)

DOD: Possibilità di visualizzare una pagina con tutti i tweet di tutte le persone che usano #leredita e provano ad indovinare la ghigliottina.

Test: Cercando per data, verificare che vengano visualizzati i tentativi del giorno.

US: Come spettatore de #leredita, voglio visualizzare, su una mappa, la posizione di tutti coloro che provano ad indovinare per conoscere la posizione dei giocatori.

Punti: 4
(4 frontend)

DOD: Possibilità di visualizzare una pagina con tutte le posizioni su una mappa dei tweet di tutte le persone che usano #leredita e provano ad indovinare la ghigliottina.

Test: Manualmente verificare che nella mappa siano presenti i marker dei tweet con la geolocalizzazione.

US: Come spettatore de #leredita, voglio visualizzare tutti coloro che indovinano la ghigliottina per sapere chi ha indovinato.

Punti: 7
(4 frontend + 3 backend)

DOD: Possibilità di visualizzare la parola del giorno assieme a tutti i vincitori che hanno indovinato.

Test: Cercando per data, verificare che venga trovata la parola del giorno.

<p>US: Come spettatore di #reazioneacatena, voglio raccogliere i tweet di chi prova a indovinare l'ultima parola, per visualizzare, in ordine temporale, tutti coloro che provano ad indovinare.</p> <p>DOD: Possibilità di visualizzare una pagina con tutti i tweet di tutte le persone che usano #reazioneacatena e provano ad indovinare la parola finale.</p> <p>Test: Cercando per data, verificare che vengano visualizzati i tentativi del giorno.</p>	<p>Punti: 2 (2 backend)</p>
---	--

<p>US: Come spettatore di #reazioneacatena, voglio visualizzare, su una mappa, la posizione di tutti coloro che provano ad indovinare per conoscere la posizione di giocatori.</p> <p>DOD: Possibilità di visualizzare una pagina con tutte le posizioni su una mappa di tweet di tutte le persone che usano #reazioneacatena e provano ad indovinare la parola finale.</p> <p>Test: Manualmente verificare che nella mappa siano presenti i marker dei tweet con la geolocalizzazione.</p>	<p>Punti: 0 Segue dalla US de #leredita</p>
--	--

<p>US: Come spettatore de #reazioneacatena, voglio visualizzare tutti coloro che indovinano l'ultima parola per sapere chi ha indovinato.</p> <p>DOD: Possibilità di visualizzare la parola del giorno assieme a tutti i vincitori che hanno indovinato.</p> <p>Test: Cercando per data, verificare che venga trovata la parola del giorno.</p>	<p>Punti: 2 (2 backend)</p>
--	--

<p>US: Come giocatore di scacchi, voglio poter muovere una pedina per fare la mia mossa.</p> <p>DOD: Possibilità di fare una mossa a scacchi e visualizzarla.</p> <p>Test: Provare a effettuare mosse valide e invalide.</p>	<p>Punti: 12 (6 frontend + 6 backend)</p>
---	--

2.3.3 Esito sprint

Lo sprint è terminato con la conclusione di tutte le user stories pianificate.

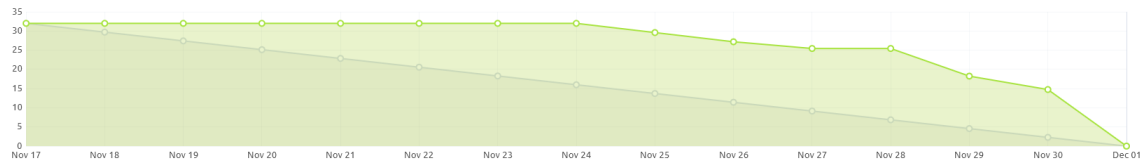


Figura 8: Burndown sprint 3

Il lavoro è stato però distribuito in maniera disomogenea, infatti, nella prima metà dello sprint non sono stati fatti progressi significativi, mentre tutto il valore è stato portato nella seconda metà dello sprint. Conseguentemente anche le ore di lavoro sono tutte distribuite nel secondo periodo dello sprint.

Il monte ore "standard" non è stato raggiunto, ma ciò era atteso in quanto durante la lo sprint planning era già stata prevista una minore quantità di lavoro.

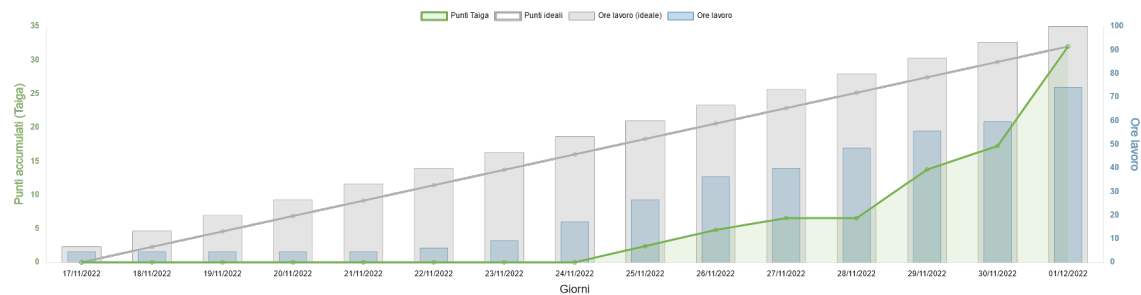


Figura 9: Progresso dei punti (asse a sinistra) e ore di lavoro (asse a destra)

2.3.4 Sprint review

Alla sprint review il cliente ha espresso le seguenti richieste:

- Revisione del grafico a barre della frequenza dei tweet, soprattutto in situazioni in cui sono presenti solo tweet di una giornata
- Aggiungere ai giochi televisivi la visualizzazione dei giocatori più vincenti in un periodo di tempo

Inoltre è stata aggiunta alle specifiche, con priorità massima, l'implementazione di funzionalità riguardanti il Fantacitorio.

2.3.5 Retrospettiva

Dalla retrospettiva sono emerse le seguenti problematiche:

- Abbiamo sottovalutato lo sprint e lavorato poco di conseguenza
- La DoD di alcune user stories risultavano ambigue

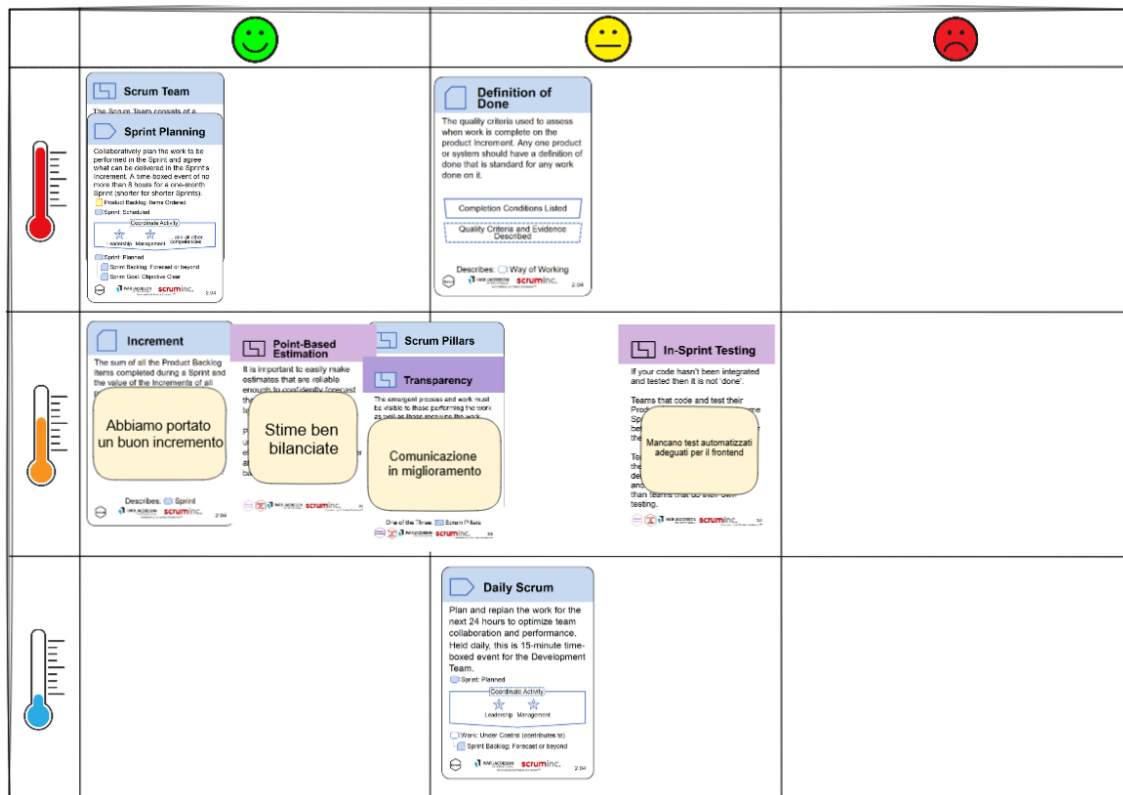


Figura 10: Retrospettiva del 02/12/2022

2.4 Sprint 4

2.4.1 Sprint goal

L'obiettivo dello sprint finale è stato quello di implementare il Fantacitorio (richiesto allo sprint precedente), concludere lo scacchi e integrare le richieste del cliente emerse dalla sprint review precedente nel prodotto finale.

In particolare sono state previste le seguenti funzionalità:

- Per il Fantacitorio:
 - Visualizzazione e modifica della classifica settimanale
 - Visualizzazione della classifica generale
 - Visualizzazione delle squadre degli altri giocatori
 - Ricerca della squadra di uno specifico utente
 - Statistiche sulla classifica
- Per lo scacchi:
 - Pubblicazione della scacchiera come tweet
 - Raccolta e selezione della mossa dell'avversario a maggioranza
- Per i giochi televisivi:
 - Visualizzazione dei più vincenti in un periodo di tempo

2.4.2 Backlog

US: Come interessato al Fantacitorio, voglio vedere e poter aggiornare una sintesi settimanale dei punteggi dei politici per vedere chi sta vincendo.	Punti: 11 (4 frontend + 7 backend)
--	--

DOD: Avere una lista di tutti i politici con i loro relativi punteggi, visualizzarne una sintesi e poter modificarla.

Test: Verificare il corretto parsing dei tweet contenente i punteggi.

US: Come interessato al Fantacitorio, voglio vedere una classifica cumulativa di punteggi di ogni politico per capire chi sta vincendo.	Punti: 5 (2 frontend + 3 backend)
--	---

DOD: Visualizzare una classifica cumulativa con i punti di ogni politico in ordine decrescente.

Test: Verificare che la classifica sia coerente con i punteggi settimanali.

US: Come interessato al Fantacitorio,
voglio sfogliare le immagini delle squadre degli altri
partecipanti
per vedere contro chi competo.

Punti: 7
(3 frontend + 4 backend)

DOD: Visualizzare, in una pagina, tutte le foto delle squadre partecipanti al Fantacitorio.

Test: Verificare manualmente che l'algoritmo sia in grado di riconoscere correttamente le immagini contenente una squadra.

US: Come interessato al Fantacitorio,
voglio poter cercare un utente e vedere se partecipa o
meno e mostrare la sua squadra
per cercare chi gioca.

Punti: 5
(2 frontend + 3 backend)

DOD: Possibilità di cercare un utente per username e controllare se possiede una squadra, in caso affermativo mostrarla.

Test: Verificare che cercando un utente che partecipa al Fantacitorio, la sua squadra sia visibile.

US: Come giocatore di scacchi,
voglio che la scacchiera venga pubblicata
in modo tale che gli utenti vedano la mossa scelta.

Punti: 3
(3 backend)

DOD: Pubblicare un tweet con la foto della scacchiera allo stato attuale.

Test: Verificare manualmente che il tweet sia stato pubblicato.

US: Come giocatore di scacchi,
voglio che gli utenti di Twitter scelgano, a maggioranza,
la mossa dell'avversario
per avanzare nella partita.

Punti: 4
(1 frontend + 3 backend)

DOD: Scelta di una mossa in base alla maggioranza dei voti presenti nei commenti del post pubblicato.

Test: Verificare che la mossa scelta sia quella proposta dal maggior numero di utenti.

US: Come spettatore de #leredita,
voglio visualizzare colui/colei che ha indovinato più volte
nel corso delle puntate,
per sapere chi è più bravo/a.

Punti: 5
(2 frontend + 3 backend)

DOD: Possibilità di visualizzare, giorno per giorno, una classifica con le persone con più parole indovinate.

Test: Verificare che gli utenti visualizzati abbiano effettivamente vinto il numero di volte indicato.

US: Come spettatore de #reazioneacatena, voglio visualizzare colui/colei che ha indovinato più volte nel corso delle puntate, per sapere chi è più bravo/a.

Punti: 1
(1 backend)

DOD: Possibilità di visualizzare, giorno per giorno, una classifica con le persone con più parole indovinate.

Test: Verificare che gli utenti visualizzati abbiano effettivamente vinto il numero di volte indicato.

US: Come interessato al Fantacitorio, voglio visualizzare delle statistiche interessanti nella classifica per vedere chi sta andando bene e chi no.

Punti: 3
(1 frontend + 2 backend)

DOD: Mostrare statistiche interessanti nella pagina della classifica, come ad esempio “best climber”, “best average” e “best single score”.

Test: Verificare che le statistiche siano in linea con la classifica.

2.4.3 Esito sprint

Lo sprint è terminato con la conclusione di tutte le user stories pianificate.

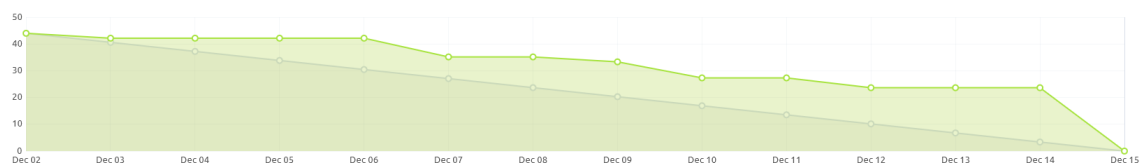


Figura 11: Burndown sprint 4

La distribuzione delle ore di lavoro è risultata disomogenea, con un'importante concentrazione all'ultimo giorno dello sprint.

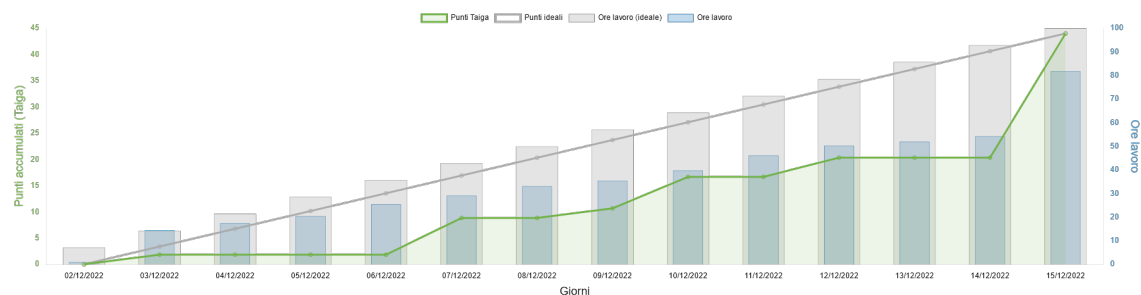


Figura 12: Progresso dei punti (asse a sinistra) e ore di lavoro (asse a destra)

2.4.4 Retrospettiva

Alla retrospettiva il team ha evidenziato i seguenti aspetti positivi:

- Una buona reazione alle modifiche del backlog
- Un incremento significativo nei test (del frontend)

È stato invece evidenziato come aspetto negativo un mal bilanciamento del lavoro, concentrato tutto alla fine dello sprint.

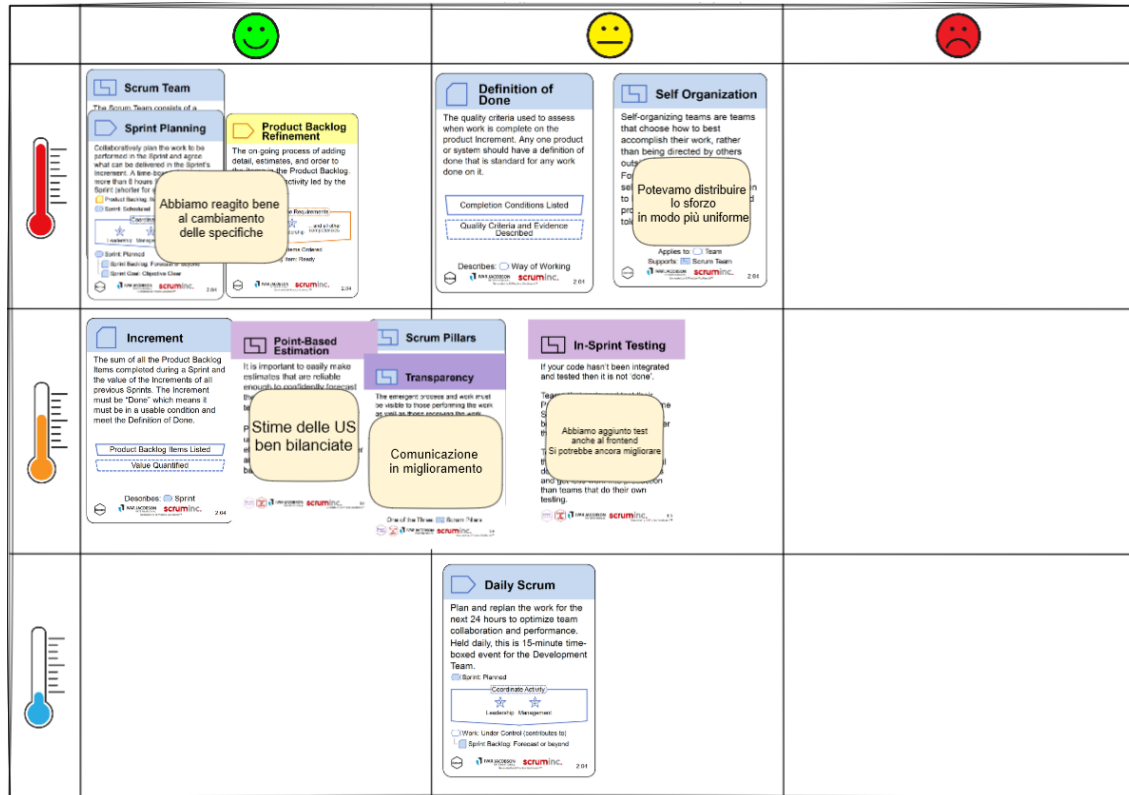


Figura 13: Retrospettiva del 02/12/2022

3 Descrizione del processo

3.1 Team building

3.1.1 Scrumble

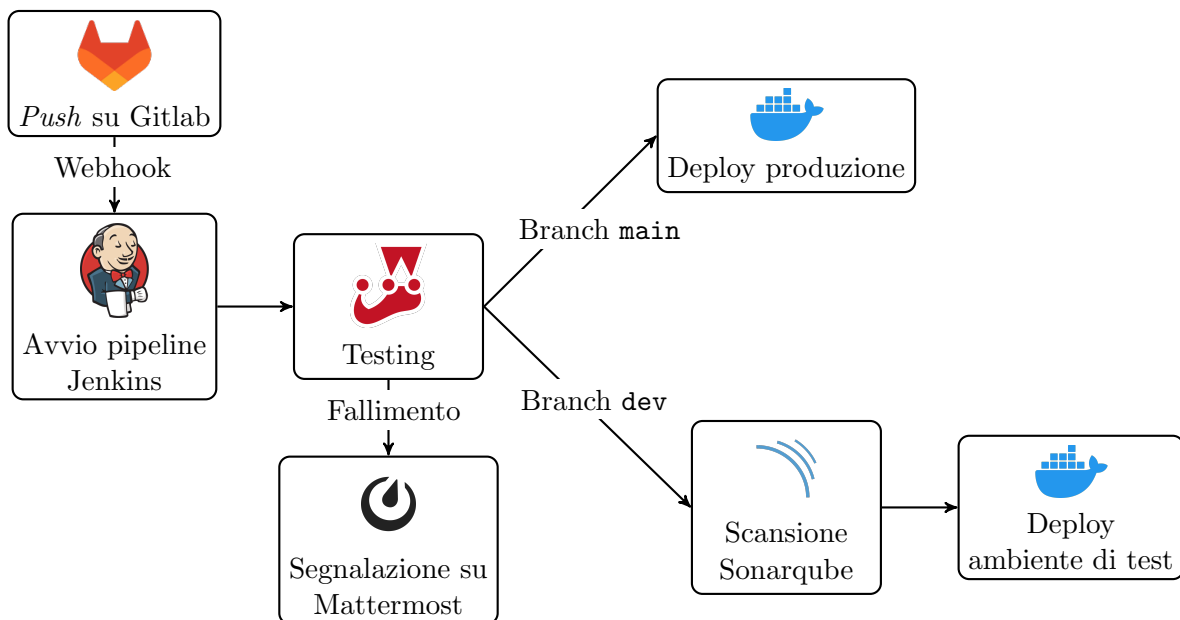
3.1.2 Escape the Boom

3.2 Gitinspector

3.3 Retrospettiva finale

3.4 Deployment

Il seguente schema rappresenta le fasi della pipeline definita per il progetto:



Sonarqube è definito solo sul branch di sviluppo (**dev**) in quanto la versione *community* è limitata alla scansione di un unico branch.

Il deploy avviene su container Docker allineati al codice del branch di riferimento. Il database risiede su un container separato.

4 Artefatti