

# Progetto di Ingegneria del software 2022

PO-Docenti (PC, MM, GS)

4 ottobre 2022

## Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Descrizione del contesto</b>	<b>2</b>
<b>3</b>	<b>Il prodotto: Le prime epiche</b>	<b>3</b>
<b>4</b>	<b>Un processo agile</b>	<b>3</b>
4.1	Struttura dei gruppi . . . . .	3
4.2	Struttura del processo . . . . .	3
4.3	Fasi del processo . . . . .	3
4.4	Inizializzazione del processo . . . . .	4
4.5	Retrospective . . . . .	4
<b>5</b>	<b>L'ambiente di sviluppo</b>	<b>4</b>
<b>6</b>	<b>Tecnologie per il prodotto</b>	<b>5</b>
<b>7</b>	<b>Valutazione</b>	<b>5</b>
<b>8</b>	<b>Link</b>	<b>5</b>

# 1 Introduzione

Vogliamo costruire un prodotto software capace di raccolta e analisi dei tweet di Twitter, specie quelli con foto e geolocalizzabili. I tweet possono riferirsi ad una persona, ad un luogo, ad un evento, ad un gioco, ad una trasmissione televisiva. In alcuni casi i tweet riguardano situazioni di emergenza. In altri possono essere usati per diffondere informazioni o per farsi pubblicità. Per questo progetto vogliamo usarli per giocare in gruppo o supportare giochi televisivi.

## 2 Descrizione del contesto

L'obiettivo generico del progetto è quello di raccogliere i tweet, organizzarli e analizzarli visualizzandoli in forma sintetica.

Lo scopo è rilevare e classificare tweet che includono particolari parole chiave esaminando i tweet stessi. La raccolta può essere storica (es. ultima settimana) o in stream in tempo reale.

L'applicazione dovrà permettere di visualizzare, consultare i tweet con certi hashtag e - sotto certe condizioni- attivare una procedura specifica. Ad esempio, il sistema dovrà essere in grado di:

- raccogliere i tweet del passato per analisi storiche (scoprirete che certe raccolte di tweet sono a pagamento: ovviamente vogliamo usare solo tweet gratis, quindi imparate a raccogliervi voi)
- classificare i tweet geolocalizzati, visualizzandoli su una mappa in una data area geografica
- classificare i tweet con una certa parola chiave, ad esempio #partitaAscacchi123
- classificare i tweet geolocalizzati di una persona specifica, per seguirne gli spostamenti
- analizzare il sentimento (*sentiment analysis*) di una serie di tweet, ad esempio un gruppo di persone che guarda e commenta una trasmissione televisiva

I tweet così raccolti e classificati potranno essere aggregati in vari modi, principalmente in modalità grafica.

In particolare si richiede di aggregare i tweet in una *dashboard* (visualizzatore) interattiva, mostrando viste coordinate con diversi dettagli dei dati:

- le posizioni di chi twitta su una mappa,
- una *term cloud* (nuvola di parole dei termini usati nei tweet),
- un diagramma a barre con il numero di tweet nell'unità di tempo,
- un grafico a torta con sentiment positivi e negativi.

**IMPORTANTE:** Ogni team può aggiungere specifiche funzioni di propria scelta al prodotto, previa approvazione del PO-docente. La proposta di funzionalità è a cura del PO del team (chiamato PO Operativo), che contatta il docente.

Siate creativi, la cosa verrà apprezzata adeguatamente in sede di valutazione finale all'esame.

Suggerimento: iscrivetevi a Twitter (se non lo siete già) e trasmettete i vostri tweet personali usando l'hashtag #IngSw2022 magari inserendo foto o commenti. Pubblicate anche tweet su luoghi o eventi o persone a vostra scelta. Attivate sulla app Twitter del vostro telefono la geolocalizzazione dei tweet (questa è una funzione volontaria di Twitter<sup>1</sup>).

---

<sup>1</sup><https://help.twitter.com/en/safety-and-security/tweet-location-settings>

### 3 Il prodotto: Le prime epiche

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 #reazionecatena* (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

### 4 Un processo agile

#### 4.1 Struttura dei gruppi

- SG<sub>1</sub>: I gruppi di progetto dovrebbero avere cinque membri. Sono tollerati gruppi con un membro in più o in meno solo se residuali
- SG<sub>2</sub>: Uno dei membri deve assumere il ruolo di Product Owner Operativo; un altro membro assumerà il ruolo di Scrum Master. Gli altri membri sono developer (sviluppatori). Tutti i membri del team, compresi POO e SM, partecipano allo sviluppo e sono collettivamente responsabili del prodotto finale.
- SG<sub>3</sub>: Il PO Operativo è il responsabile del prodotto entro il team, e partecipa allo sviluppo. Si raccorda con i PO-docenti, che sono stakeholder capaci di (ri)definire i requisiti del prodotto. Il PO è anche sviluppatore.
- SG<sub>4</sub>: Lo Scrum Master è il facilitatore. Ha inoltre il compito di coordinatore di processo entro il team (e quindi si occupa dei documenti di processo, in particolare del rapporto finale, che comunque verrà firmato da tutti i membri del team). Anche lo Scrum Master può partecipare allo sviluppo.

#### 4.2 Struttura del processo

- SP<sub>1</sub>: Il processo si svolge su sprint di due settimane ciascuno, a partire dal 5 ottobre.
- SP<sub>2</sub>: Il primo sprint si chiama *sprint zero* e serve per la fase preliminare (team forming, team building) ed a inizializzare l'ambiente di sviluppo.

#### 4.3 Fasi del processo

**Fase preliminare - sprint zero** a) Team forming tramite Trello b) Addestramento a Scrum del gruppo mediante uso del gioco Scrumble (vedere link alla fine di questo documento), con raccolta degli indicatori di valutazione del team da parte dello Scrum Master. Una partita a Scrumble si può giocare sia in presenza sia con Teams e richiede circa un paio di ore.

**Fase di esecuzione** Il team esegue il progetto con sprint (numerati da 1 in poi) della durata di due settimane ciascuno. Al termine di ciascuno sprint il team dovrà preparare una demo dello stato di avanzamento del prodotto (*Sprint review*), una riflessione sul processo (*retrospettiva*) usando le carte Essence, e una sessione di *backlog grooming* cioè una revisione del backlog dopo una riunione coi PO docenti.

IMPORTANTE: all'inizio di ciascuno sprint il PO-Docente potrà fornire nuove user story, che dovranno essere aggiunte alle altre, eventualmente anche modificandole.

**Fase conclusiva** L'ultimo sprint include la preparazione del rapporto finale a cura del team, inclusa l'autovalutazione del team (verranno date indicazioni sul rapporto finale e sull'autovalutazione entro la fine del corso). Si richiede la preparazione di un breve video con audio (max 3 min.) di presentazione del prodotto sviluppato, che mostri anche un'analisi "interessante" eseguita con il prodotto stesso. La consegna di tutti i documenti e del prodotto avviene sul wiki di Taiga del team oppure sul sistema di versionamento prescelto (es. GitLab).

#### 4.4 Inizializzazione del processo

Allo scopo di mostrare la flessibilità dell'approccio agile, ciascun team decide e inizia lo sviluppo con un proprio insieme di user story che potrà essere integrato in corso d'opera con altre, richieste dal PO.

**Esempi** (lista non esaustiva): aggiungere un elemento di filtro/selezione dei tweet, come la possibilità di selezionare una finestra temporale, in modo che l'utente del sistema possa selezionare i tweet solo di un determinato periodo; aggiungere la possibilità di cliccare su una delle parole della tag cloud in modo da filtrare i dati sulla base di tale parola chiave e di vedere quindi la mappa aggiornata di conseguenza; aggiungere altre sintesi grafiche di data analytics, quale un istogramma che mostri la frequenza con cui è comparsa una parola chiave nella finestra temporale prescelta.

#### 4.5 Retrospettive

Ogni sprint si conclude con una retrospettiva, che è una discussione e autovalutazione del team sullo stato del processo seguito. Vi avvarrete delle carte Essence per definire il risultato di ciascuna retrospettiva. Le carte Essence [3] per le retrospettive si trovano al link di cui sotto. Di ogni retrospettiva lo Scrum Master redige un verbale in forma succinta, che verrà allegato alla relazione finale.

### 5 L'ambiente di sviluppo

L'uso dei servizi dell'ambiente di sviluppo CAS [1] è obbligatorio. Il nostro scopo è di fare una esperienza di sviluppo "estrema", cioè totalmente basata su tecnologie open source [4] deployate su hardware in nostro controllo. Inoltre vogliamo raccogliere dati di processo grazie alle funzionalità presenti nell'ambiente CAS [2]. L'ambiente CAS contiene i seguenti servizi, tutti open source:

1. Taiga (simile a Trello) per il project management e per raccogliere alcuni documenti nel suo wiki
2. Mattermost (simile a Slack) per le comunicazioni tra gli sviluppatori
3. SonarQube per l'analisi statica del codice
4. gitlab per il controllo delle versioni e della configurazione
5. jenkins per il testing e la *Continuous Integration*
6. logger server, per raccogliere dati di sviluppo direttamente dai client IDE arricchiti con plugin (disponibili plugin per Eclipse, Atom, IntelliJ, Visio).

Questi servizi (o servizi similari, open source e previa autorizzazione dei PO-docenti) andranno configurati su una vostra macchina (o su un cloud privato, se ne avete disponibilità). Per ogni servizio open source installato su vostri spazi, correttamente utilizzato nel progetto e documentato nella relazione finale avrete punti aggiuntivi. In prima istanza vi chiediamo di fare un breve video (max tre minuti) non appena avrete installato i servizi su vostre macchine; il video dimostrerà che avete installato i servizi come richiesto; il video andrà fatto entro lo sprint zero.

## 6 Tecnologie per il prodotto

Per lo sviluppo potete usare Java, Javascript, Python o altre tecnologie, a vostra scelta. Si richiede che il deployment sia comunque basato su docker; si consiglia l'uso degli spazi Web dipartimentali. Per quanto possibile TUTTE le tecnologie utilizzate dovrebbero essere open source.

## 7 Valutazione

Al team viene dato un voto uguale per tutti i membri basato su: demo del prodotto e qualità del rapporto finale. La durata del processo sarà un fattore di valutazione influente sul voto (più tardi consegnate minore sarà il voto).

Altre informazioni verranno date a lezione.

## 8 Link

Twitter <https://developer.twitter.com/en/docs>

Codice di CAS <https://github.com/elPeron>

Tweet-Tracker <https://github.com/Zacomo/Tweet-Tracker>

Scrumble <http://scrumble.pyxis-tech.com>

Essence introduzione <http://semat.org/essence-user-guide>

Carte Essence <https://practicelibrary.ivarjacobson.com/start>

## Riferimenti bibliografici

- [1] P. Ciancarini, M. Missiroli, F. Poggi, and D. Russo. An open source environment for an agile development model. In *IFIP International Conference on Open Source Systems*, volume 582 of *Advances in Information and Communication Technology*, pages 148–162. Springer, 2020.
- [2] P. Ciancarini, M. Missiroli, and S. Zani. Empirical evaluation of agile teamwork. In *International Conference on the Quality of Information and Communications Technology*, pages 141–155. Springer, 2021.
- [3] I. Jacobson, H. Lawson, P. Ng, P. McMahon, and M. Goedicke. *The Essentials of Modern Software Engineering*. ACM Books. Morgan & Claypool Publishers, 2019.
- [4] P. Marzolo, M. Guazzaloca, and P. Ciancarini. Extreme development as a Means of Learning Agile. In *Proc. 1st Int. Conf. on the Frontiers of Software Engineering*, volume 1523 of *Communications in Computer and Information Science*, pages 158–175. Springer, 2021.