

Relazione Progetto Ingegneria del Software 2

Machine Learning for Software Engineering

Christian Santapaola 0294464

Università di Roma Tor Vergata

February 9, 2022

- github.com/christiansantapaola
- [sonarcloud](https://sonarcloud.io)

- Introduzione
- Descrizione del problema
 - Issue Tracking System: Jira
 - Version Control System: Git
 - Calcolo della metrica
- Output
- Analisi dei risultati

- Lo scopo di questo progetto è stato di analizzare la stabilità di una metrica software di un sistema software, tramite l'ausilio di il sistema di Issue Tracking jira e il sistema di Version controll git.
- La Metrica da valutare è il numero di bug risolti per unità di tempo.
- Il progetto preso in considerazione è stato S2Graph.
- L'obiettivo finale dell'analisi è la generazione del *Process Control Chart*.

Descrizione del Problema

- Dobbiamo calcolare il numero di bug risolti.
- Dal sistema di issue tracking Jira dobbiamo prendere i dati riguardanti i Ticket dei bug con relative informazioni di interesse:
 - Stato corrente del Bug.
 - ID Del Ticket.
- Infine dobbiamo cercare il commit con la risoluzione del bug, questo commit avra nel suo messaggio l'ID del ticket recuperato in precedenza; dal commit vogliamo recuperare la data di commit.
- ottenute le informazioni possiamo calcolare il *Process Controll Chart*.

Jira - Ottenere i Ticket

- Jira offre un servizio web per formulare query sui ticket e ottenere i ticket richiesti in formato JSON.
- Una nota dolente di questo progetto è che i bug risolti non vengono taggati come FIXED o RESOLVED, ma come DONE, inoltre i BUG risolti non vengono considerati come chiusi.
- Conseguenza di ciò è stata la necessità di modificare la query affinché trovasse i ticket richiesti. Seconda nota dolente è stato il numero di ticket riguardanti i bug sistemati, che sono molto pochi e molto sparsi nel tempo.

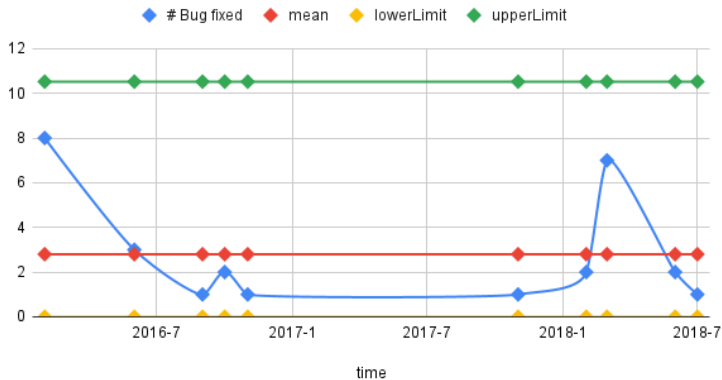
Git - Ottenere la Committer Date

- Presi gli ID dei ticket, è stato possibile recuperare i Commit con la bugfix tramite un operazione di grep sulla commit history, cercando i commit contenenti l'ID del ticket.
- Per ogni Ticket è stata considerata come data di chiusura del ticket la Committer Date, perché questa data risulta più accurata come data di chiusura del ticket.
- Per analizzare la git history è stata utilizzata la libreria JGIT, un implementazione nativa in java di GIT, le motivazioni sono state le seguenti:
 - Avere i dati necessari nativamente come oggetti Java.
 - Rendere il programma non dipendente da un installazione git presente sul sistema.

Calcolo della metrica

- Per calcolare la metrica richiesta sono stati contati il numero di Ticket di risoluzione di bug aggregati per la coppia (anno,mese) in base alla commiter date.
- Il *Process Controll Chart* ha lo scopo di valutare la l'andamento di un processo nel tempo, così da poter effettuare uno studio di tipo statistico sul processo, andando a valutare se il processo supera determinati valori soglia nel suo andamento.
- in particolare i valori soglia valutati sono stati chiamato P_t il nostro processo:
 - La media $E[P_t]$
 - La soglia superiore calcolata come $\text{UpperLimit} = E[P_t] + 3\sqrt{\text{Var}(P_t)}$
 - La soglia inferiore calcolata come $\text{LowerLimit} = E[P_t] - 3\sqrt{\text{Var}(P_t)}$
 - Sia la media che la deviazione standard sono state calcolate usando la media campionaria e la varianza campionaria.

ISW2 - Deliverable 1 - S2GRAPH Bug fixed



- Dal grafico si può notare la sparsità dei risultati, il processo presenta periodi morti uniti a periodi con alti picchi di risoluzione, questo evidenzia una saltuarietà nel lavoro di questo progetto software.
- Tuttavia nessuno dei picchi supera la soglia massima consentita.