TOR VERGATA
UNIVERSITÀ
DEGLI STUDI
DI ROMA

Macroarea di Ingegneria
Corso di Laurea Magistrale in Ingegneria Informatica

Ingegneria del Software II (9 CFU) A.A. 2020/2021
Docenti Proff. Davide FALESSI, Guglielmo DE ANGELIS, Giuseppe F. CALAVARO

Modulo *Machine Learning for Software Engineering*

Deliverable I

Massimo STANZIONE
Matr. 0304936

Indice

Introduzione – Contesto

- **Obiettivo**
  Presentazione dei risultati ottenuti dall'analisi della stabilità del numero di ticket risolti nel durare del ciclo di vita di un progetto open-source.
- **Progetto in analisi**
  Apache S2Graph, software per il processamento di informazioni mantenute su basi di dati con strutture a grafo.
- **Versione scelta**
  La più recente disponibile in GitHub (gennaio 2019), non essendo necessario effettuare misurazioni di classi e vista la limitata disponibilità di versioni, la cui più recente risale al 2017.
- **Ambienti di sviluppo**
  - Eclipse 2020-09 su Debian 9
  - IntelliJ IDEA 2021.2.3 su Linux Mint 20.2

Progettazione – Modellazione del problema

Il problema della analisi dei ticket è stato modellato ed affrontato **per fasi successive**, mediante l'ausilio della repository del progetto disponibile su *GitHub*, dello strumento di issue tracking *JIRA* e del programma di calcolo elettronico *LibreOffice Calc*.

Le fasi previste ed eseguite sono state le seguenti:

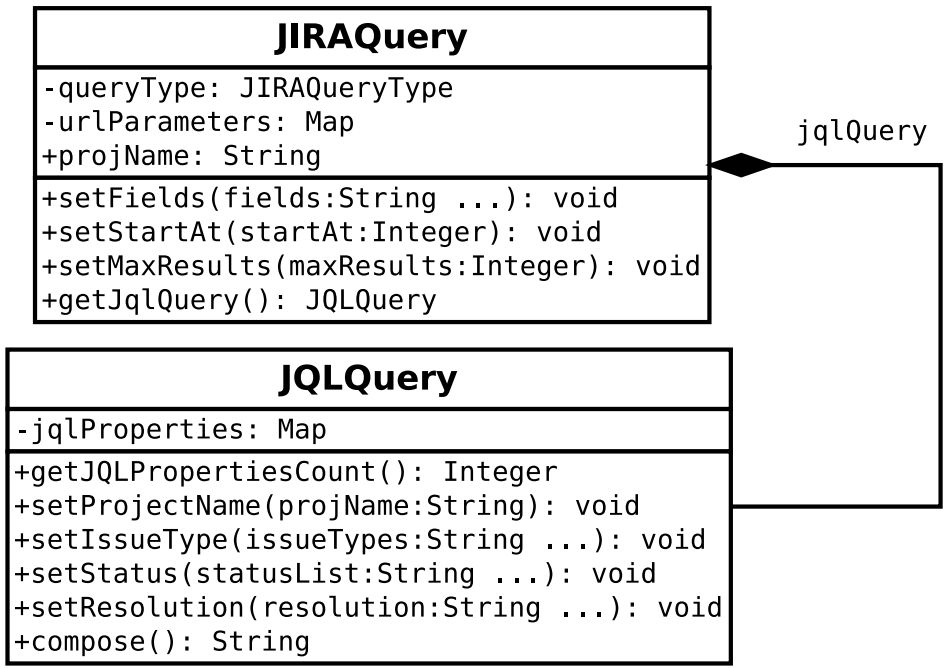
1. Generazione locale di una ***working copy*** del progetto;
2. Fetching dei **commit** del progetto;
3. Analisi dei **ticket** relativi a bug di tipo *"fixed"*;
4. **Confronto** dei commit relativi ai ticket prelevati;
5. Predisposizione di un ***Process Control Chart***;
6. **Analisi dei dati** e conclusioni.

Progettazione – Interfacciamento con JIRA

Le informazioni relative ai ticket sono state prelevate dal sistema di issue tracking JIRA, mediante le API da esso esposte.

Dal punto di vista implementativo, tale operazione è stata effettuata mediante predisposizione di **due apposite classi**, `JIRAQuery` e `JQLQuery`, tramite le quali sono state manipolate le queries in linguaggio JQL (JIRA Query Language) impiegate per il prelievo delle informazioni.

I risultati, in formato JSON, sono poi stati manipolati con l'ausilio di una classe similmente predisposta, `JSONHandler`.



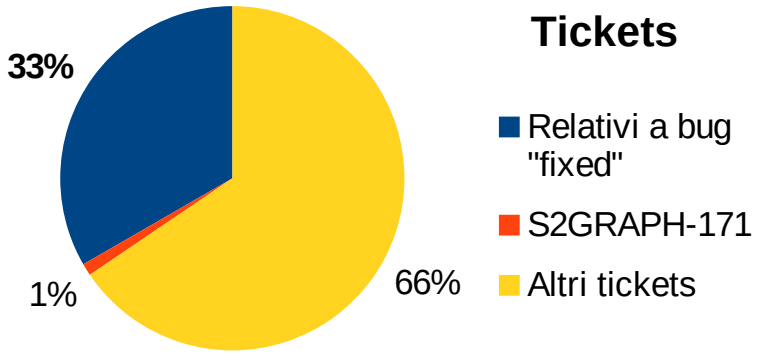
Dataset – Analisi

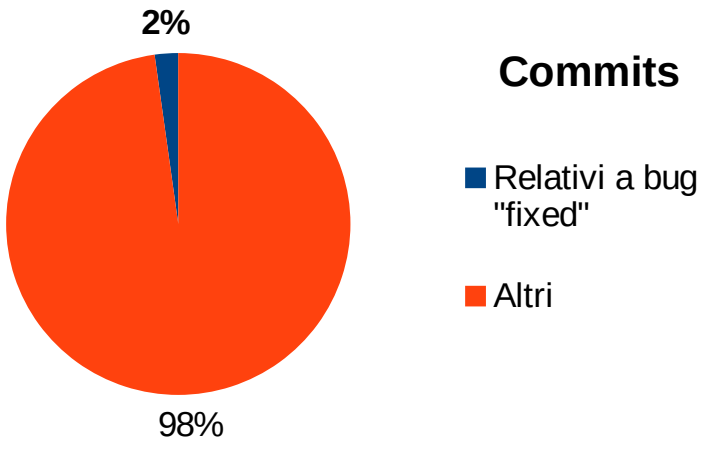
Secondo quanto ricavato, i dati che saranno analizzati sono i seguenti:

- **87** tickets
  - dei quali **30** (33.1%) per ticket **relativi a bug di tipo “fixed”**
    - di cui **1** (S2GRAPH-171) **scartato** in quanto privo di commit assegnati
- **1681** commits
  - dei quali **37** (2.2%) **relativi ai ticket considerati**.

La **probabilità di linkage** è pari a 0.96.

Si esamineranno in seguito le motivazioni della scarsità di informazioni disponibili.





Dataset – Manipolazione

Allo scopo di predisporre adeguatamente i dati da analizzare, per ogni ticket valido è stata realizzata una lista contenente tutti i commit ad esso relativi, e di essi è stato considerato il **commit con la data più recente**, da considerarsi come **data di fix** del bug cui il ticket fa riferimento.

Le date così elaborate e raccolte sono state **raggruppate per mese**, ed esportate su un file CSV (Comma Separated Values) per poter essere poste come oggetto di analisi.

L'analisi è proseguita, con l'utilizzo del software Calc, con il **calcolo delle medie e delle deviazioni standard**, propedeutiche ai limiti di controllo e alla linea di media del *Process Control Chart.*

	A	B
1	Date	Occurrences
2	feb 2016	8
3	mag 2016	1
4	giu 2016	2
5	set 2016	1
6	ott 2016	2
7	nov 2016	1
8	nov 2017	1
9	feb 2018	2
10	mar 2018	7
11	giu 2018	3
12	lug 2018	1

Process Control Chart

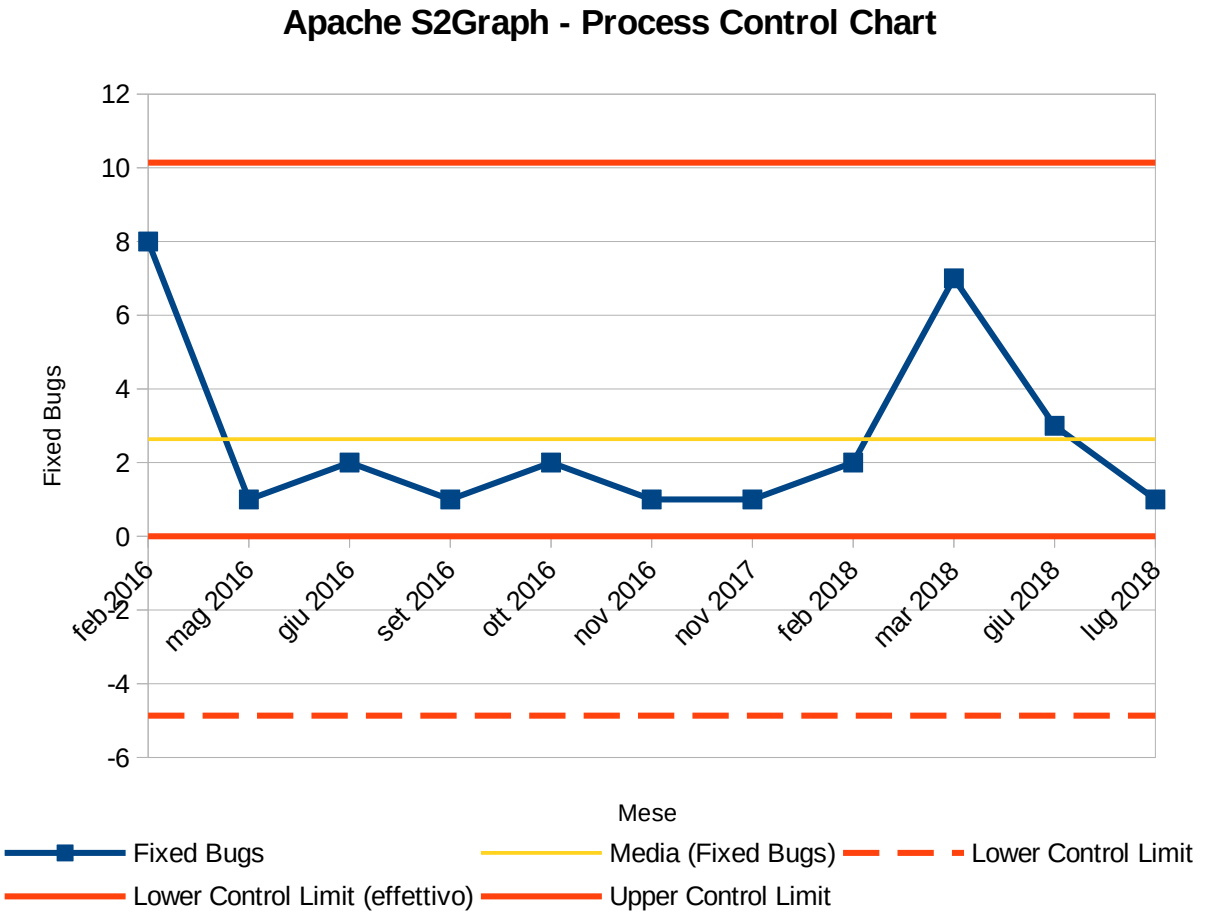
I **limiti di controllo** sono definiti come:

- $UCL = \mu + 3\sigma$
- $LCL = \begin{cases} \mu - 3\sigma & se \quad \mu - 3\sigma > 0 \\ 0 & altrimenti \end{cases}$

Si evidenzia la presenza di un **numero maggiore di bug risolti** nei mesi di febbraio 2016 e di marzo 2018.

Ad eccezione di questi due casi e del mese di giugno 2018, il numero di bug risolti si è sempre mantenuto **al di sotto del valore medio**.

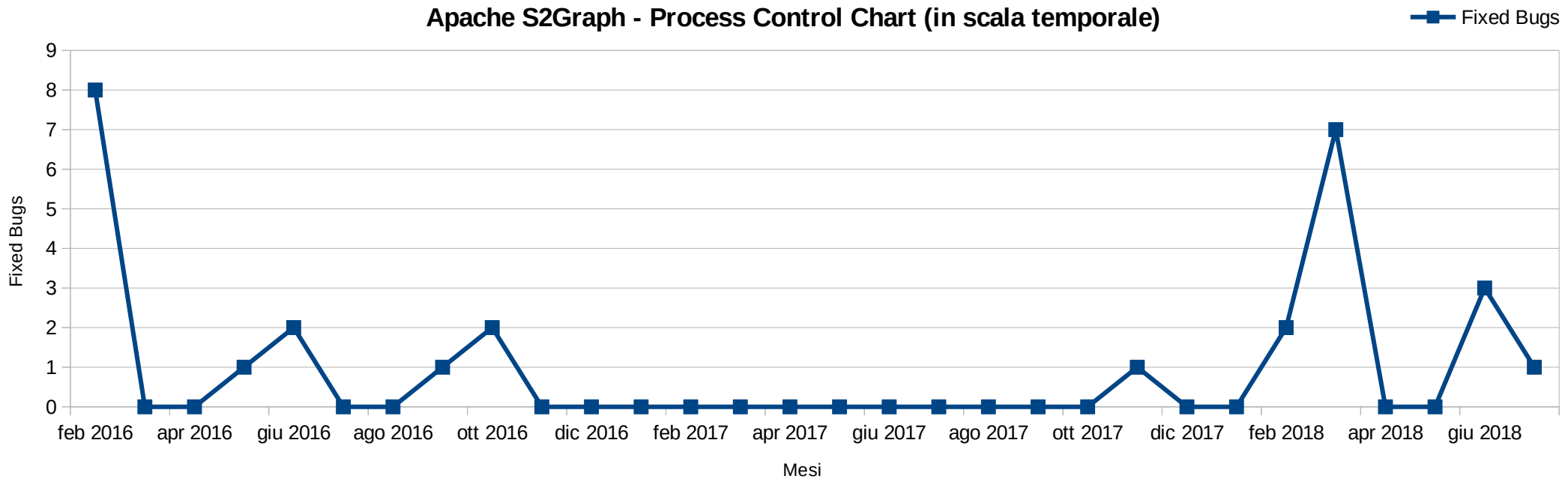
I valori rientrano, in ogni caso, **all'interno dei limiti di controllo**.



Discussione dei risultati

Per comodità di analisi è riportato di seguito il Process Control Chart, ma **in scala temporale** e privato dei limiti di controllo.

Si nota la presenza di un periodo di circa un anno (nov 2016 – ott 2017) in cui **non è stato risolto alcun bug** relativo ai ticket considerati.



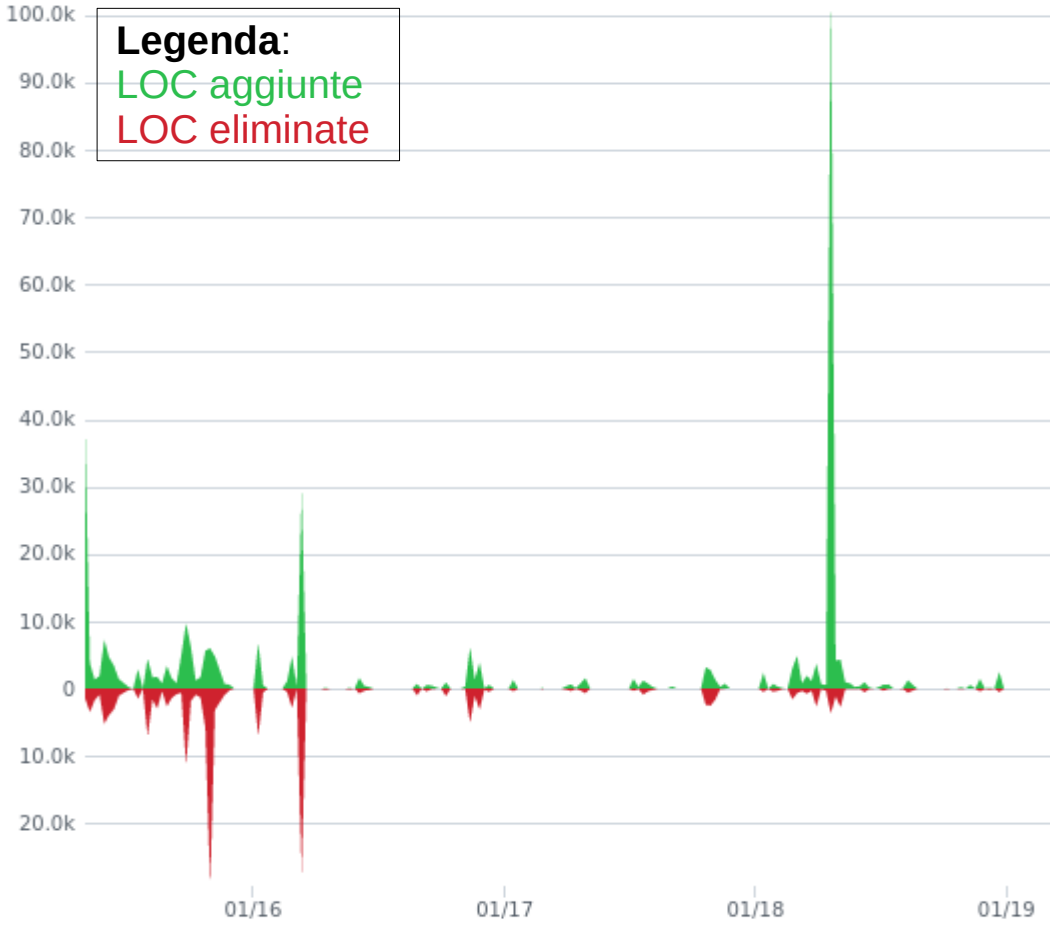
Discussione dei risultati

Nel grafico è riportato il ***Code frequency graph*** del progetto, disponibile su GitHub.

La scarsità di informazioni disponibili e il periodo temporale con assenza di bug risolti sono dovuti alla **discontinuità del progetto**; è possibile inoltre notare che uno dei periodi con minore attività è coincidente con quello considerato nella slide precedente.

Il progetto è inoltre **non più mantenuto** da gennaio 2019.

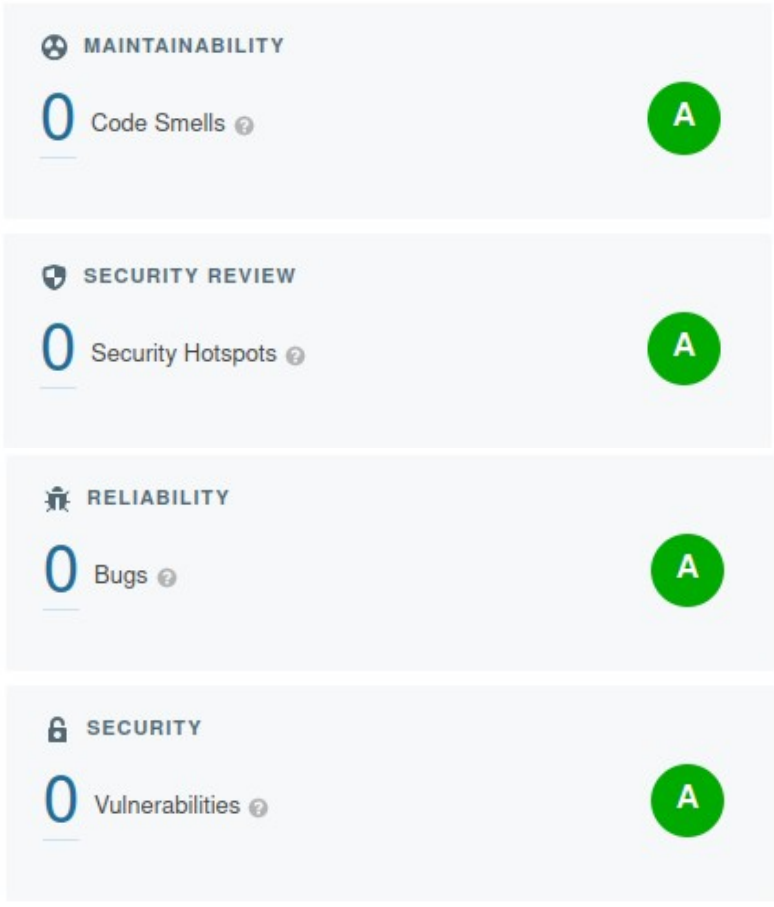
A parziale compensazione, l'**elevata probablità di linkage**, pari a 0.96, ha fatto sì che non siano state disperse ulteriori informazioni rilevanti per i ticket.



Analisi del codice

Il codice è stato analizzato mediante *SonarCloud*, e l'analisi è stata **inclusa nel ciclo di build del progetto**, utilizzando *Maven* per la gestione delle dipendenze e *CircleCI* come strumento di CI/CD.

In seguito ad un processo di revisione del codice i bug, le code smells ed i problemi di sicurezza sono stati azzerati.



Riferimenti

- Sito del progetto S2Graph:
  https://incubator.apache.org/projects/s2graph.html
- Repository S2Graph:
  https://github.com/apache/incubator-s2graph
- Pagina JIRA per S2Graph:
  https://issues.apache.org/jira/projects/S2GRAPH/issues
- Repository della deliverable:
  https://github.com/massimostanzione/isw2-deliverable1
- Analisi *SonarCloud*:
  https://sonarcloud.io/summary/overall?id=massimostanzione_isw2-deliverable1