

REPORT DELIVERABLE 1

Nome : Chiacchia Matteo

Matricola : 0300177

Email : matteoch99@gmail.com

Repository **Github**: <https://github.com/chiacchius/deliverable1>

SonarCloud: https://sonarcloud.io/dashboard?id=chiacchius_deliverable1

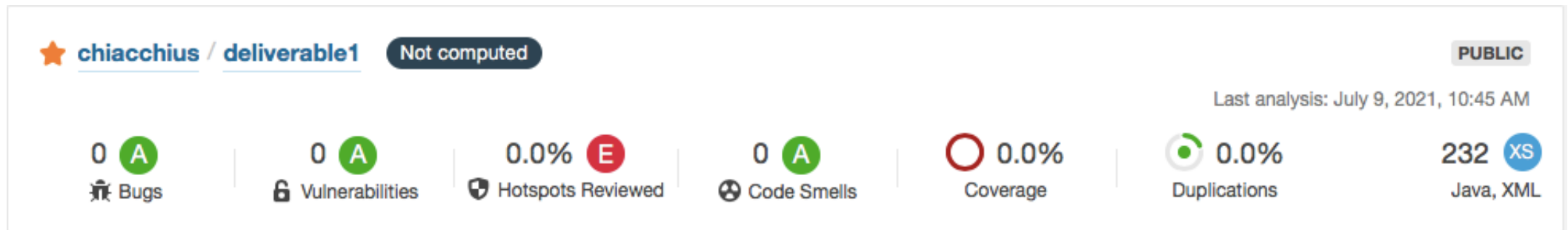
**Misurare la stabilità di un
attributo di progetto**

Note preliminari

- Descrizione del processo di *analisi* e *studio* effettuato su un progetto *open source* di **Apache**
- Risultati ottenuti tramite *automazione* fornita mediante *codice Java*, di conseguenza ottenibili per un progetto qualsiasi.
- Applicazione *hostata* sulla piattaforma di **GitHub**
- Analisi del codice effettuato tramite **SonarCloud**

Tecnologie e software utilizzati

- **Java 15.0.1** come *enviroment* di sviluppo
- **Eclipse IDE** come IDE
- **Github** e **Jira** per il raccoglimento di informazioni
- **Excel** per manipolazione ed estrapolazione dei dati
- **Github** e **git** come sistema di **Remote Repository** e **Versioning Control**
- **SonarCloud** come strumento di analisi del codice



SonarCloud values

Descrizione iniziale

- **Misurazione della stabilità** di un **attributo** di **progetto**, in particolare il numero di **fixed ticket per ogni mese** dalla data di creazione del progetto fino a oggi.
- Progetto analizzato: **STDCXX**

Perché è importante trattare la stabilità di un attributo di progetto?

- **Monitorare, controllare e predire** il comportamento del progetto durante il **processo di sviluppo**
- L'attività di misurare permette di **prendere decisioni** basate su **evidenze oggettive**, e quindi assicurare una *certa qualità* sia per quanto riguarda il progetto che per quanto riguarda il **processo** che ha come prodotto il sistema software.

Analisi dell'applicativo

- Per il raccoglimento dei **ticket** relativi a **bug fixati/risolti** viene utilizzato il sistema di **Ticket Tracking** offerto da **JIRA**.
 - Viene effettuata un'interrogazione ai server di JIRA, la quale ritorna un file ".json" da cui si può fare *parsing* per ottenere le informazioni volute.
- Per l'ottenimento delle informazioni sulla data dell'effettivo *fix* di un determinato *ticket* viene utilizzato **Github**.
 - Si ottiene l'ultimo *commit* relativo a un ticket per ottenere la data di risoluzione di quest'ultimo.

Analisi dell'applicativo

- **Passo 1:** Si clona il progetto che si vuole analizzare da *Github*.
 - L'interfaccia con **GitHub** è stata creata usando l'**API JGit**
- **Passo 2 :** Si ottengono tutti i *commits* effettuati e si salvano in un array.
- **Passo 3 :** Si ricavano, da *Jira*, tutti i ticket relativi ai *ticket* fixati/*risolti*.
- **Passo 4 :** Si confrontano tutte *key* dei ticket con il *fullMessage* del *commit*. Quando si trova l'ultimo *commit* temporale relativo al *ticket* si ricava la data che sarà la *fix date*.
- **Passo 5 :** Si crea il *Csv* associando a ogni ticket la *fix date*.

Indirizzi utilizzati per i ricavare i dati:

Retrieve ticket :

<https://issues.apache.org/jira/rest/api/2/search?jql=project=%22projName%22AND%22resolution%22=%22fixed%22&fields=key,resolutiondate,versions,created&startAt=start&maxResults=finish>

Repository :

<https://github.com/apache/projectName>

Perché utilizzare *Github* e *Jira*?

- Consapevolezza che la data del *fixed ticket*, riportata come campo per ogni ticket su *Jira*, potrebbe non essere coerente con l'effettiva data in cui questo è stato risolto.
- Per avere dati decisamente più precisi è necessario ispezionare i *commit* effettuati.

Analisi del file .csv

- Dati ottenuti vengono *formattati* e *salvati* all'interno di file **.csv**
- Calcolo di statistiche relative ai dati ottenuti
 - **Media** dei fixed ticket
 - **Deviazione standard** usata per il computo del **lower/upper limit**
 - **Percentuale** di **Ticket** non linkati a nessun *commit* su *GitHub*
- Le scelte che sono state fatte relativamente al **raccoglimento** e **interpretazione** dei dati
 - ticket viene **scartato** se non presente un commit relativo su *GitHub*
 - operazioni di analisi effettuate sul foglio **Excel**
 - Se il **lower limit** è sotto lo **zero**, allora viene automaticamente impostato a 0
- Se uno dei punti del grafico non è nell'intervallo [*lowerLimit*, *upperLimit*] il processo è considerato **instabile**.

Analisi dei dati

	2005	2006	2007	2008	2009	2010	2011	2012
1	0	14	8	10	0	0	0	1
2	0	9	10	14	0	0	0	3
3	0	9	23	16	0	0	1	0
4	0	2	13	62	1	0	0	0
5	0	2	11	27	2	0	0	0
6	0	11	18	19	0	0	0	1
7	1	7	14	12	0	1	0	0
8	1	5	15	10	1	0	0	0
9	13	1	32	6	1	0	0	0
10	12	4	39	2	0	0	0	0
11	10	3	19	5	0	0	0	0
12	8	1	12	1	0	0	0	0

Tabella dei *fixed ticket* ogni mese di ogni anno

average	5,447916667
std deviation	9,630947695

Media e **deviazione standard** del numero di *fixed ticket* al mese

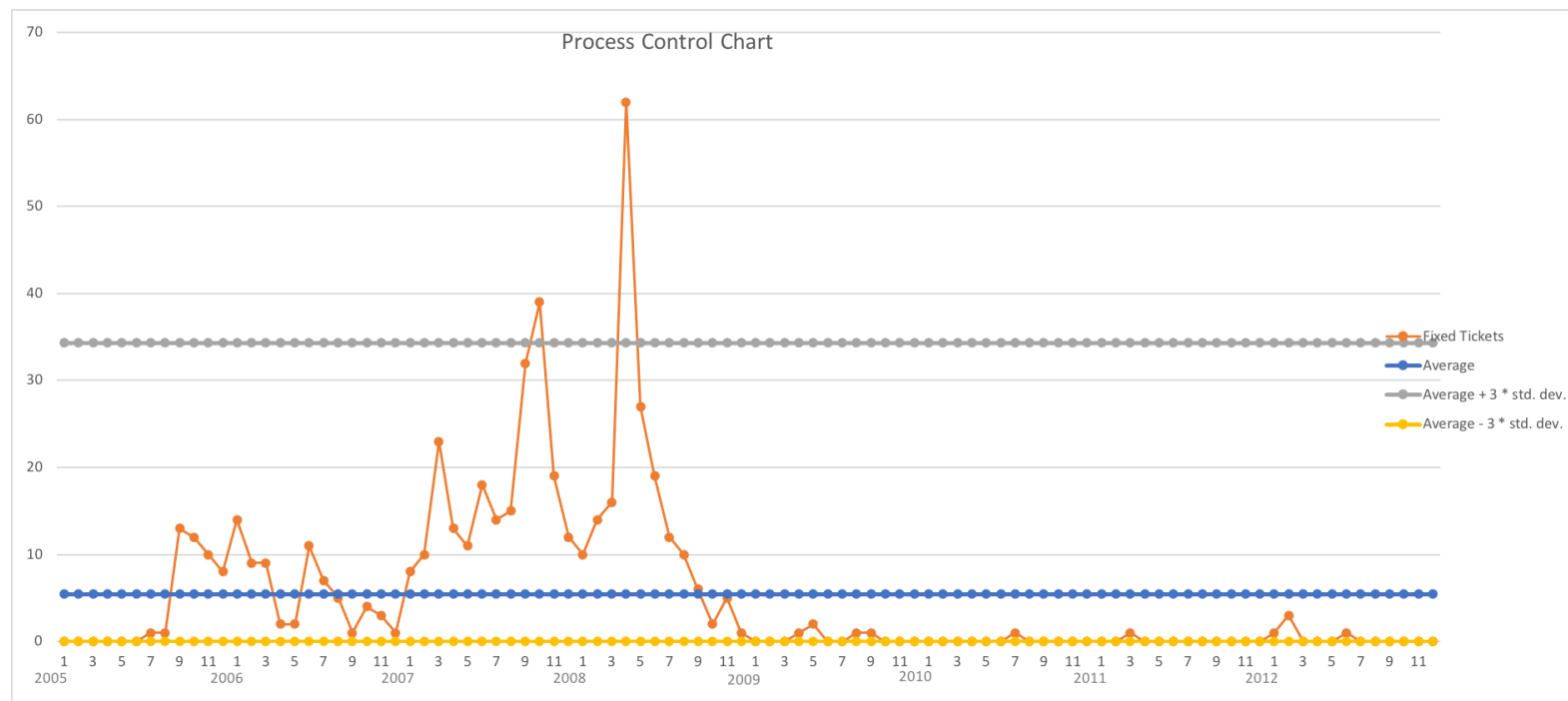
average +3*std dev	34,34075975
average-3*std dev	-23,44492642

Lower e **upper limit** calcolati

Percentage of tickets without fixDate	14%
---------------------------------------	-----

Percentuale ticket senza commit

Analisi dei dati



Il grafico dell'andamento di **STDCXX**. Il *lower limit* è stato portato da un valore negativo a 0 poiché non ha senso parlare di valori negativi quando si parla di conteggi

Analisi dei dati

- E' evidente come con il tempo il processo di sviluppo di **STDCXX** sia stato abbandonato per quanto riguarda il numero di *fixed ticket*.
 - Da inizio 2009 il numero di *fixed ticket* è prossimo allo 0.
- Nella prima metà dello sviluppo del progetto il numero di *fixed ticket* aumenta notevolmente.
 - A ottobre 2007 e aprile 2008 si nota che il numero di *fixed ticket* è sopra l'*upper bound* calcolato.
 - Si deduce un notevole incremento della produttività.
 - Progetto **immaturo** e **instabile**.
- Percentuale di *Ticket* non linkata equivalente al 14%.
 - Percentuale non troppo elevata.
 - La maggior parte dei *ticket* su **Jira** sono stati trovati nel *commit*.