



Deliverable 1

PROCESS CONTROL CHART

MARTINA SALVATI 0292307

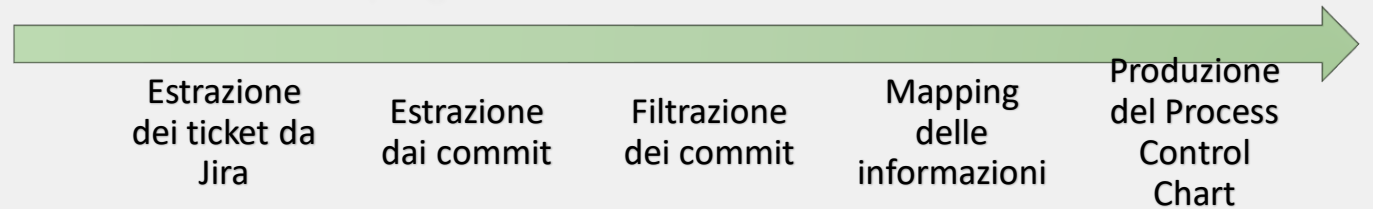
Contenuti nelle diapositive

Introduzione e obiettivi del progetto

- Introduzione, obiettivi e tecnologie utilizzate

Progettazione

- Introduzione alla progettazione



Risultati

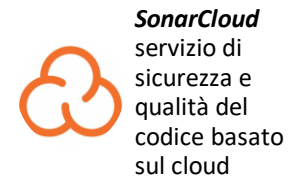
- Link al file prodotto

Riferimenti

- Sonarcloud
- Github

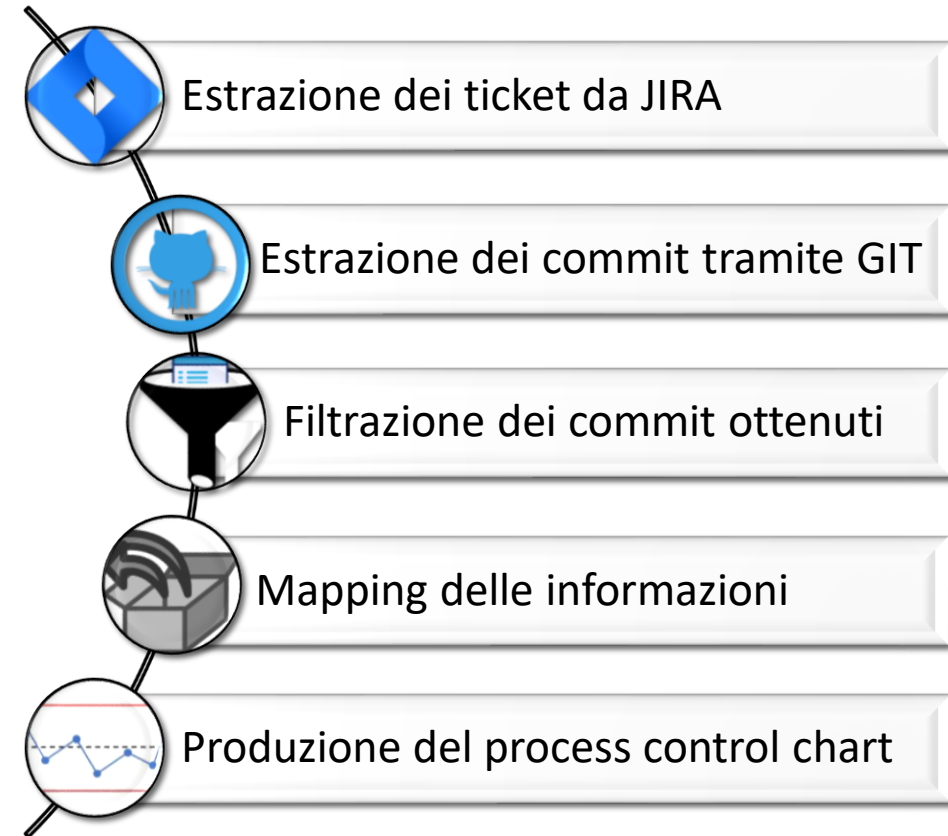
Introduzione e obiettivi del progetto

- ❖ Obiettivo del progetto è la realizzazione di un *Process Control Chart* per monitorare le variazioni nel tempo del *numero di Bug Fixed* del progetto open-source S2Graph
- ❖ Il process control chart è un grafico utile per analizzare le prestazioni di un processo nel tempo, tracciando i punti dati (#bugfixed) , limiti di controllo (superiore e inferiore) e linea centrale (media). Si può considerare uno strumento predittivo per un monitoraggio regolare di un processo.
- ❖ Gli strumenti e tecnologie utilizzati per il progetto:



Introduzione alla progettazione

- ❖ Per lo svolgimento del progetto è stato implementato un progetto in Java
- ❖ Le varie fasi del progetto prevedono l'estrazione delle informazioni dal servizio JIRA e GIT
- ❖ Le informazioni ottenute, relative ai ticket da JIRA e ai commit ricavati da GIT sono state mappate per creare un dataset di informazioni utili al fine dello svolgimento del progetto
- ❖ Una volta ottenuti i dati, sono stati trasformati nelle tabelle excel necessarie per la produzione del PCC



Progettazione : Estrazione dei ticket da JIRA

- ❖ Per l'estrazione dei ticket da JIRA è stata utilizzata la stringa di ricerca :

```
134     String url = "https://issues.apache.org/jira/rest/api/2/search?jql=project=%22"
135           + projName + "%22AND%22issueType%22=%22Bug%22AND(%22status%22=%22done%22OR"
136           + "%22status%22=%22resolved%22)&fields=key,resolutiondate,versions,created&startAt="
137           + i.toString() + "&maxResults=" + j.toString();
```

- ❖ Questa ha permesso di ottenere un oggetto di tipo JSON con tutte le informazioni necessarie dei vari ticket filtrati con lo specifico issueType e Status
- ❖ I risultati ottenuti sono stati salvati in un file di tipo csv

```
S2GRAPH-186,2018-03-21T09:46:57.000+0000
S2GRAPH-181,2018-02-26T09:58:34.000+0000
S2GRAPH-179,2018-02-26T05:04:28.000+0000
S2GRAPH-178,2018-02-26T02:18:16.000+0000
S2GRAPH-176,2018-02-13T09:55:28.000+0000
S2GRAPH-171,2017-11-14T08:46:02.000+0000
S2GRAPH-168,2017-10-26T12:49:33.000+0000
S2GRAPH-126,2016-11-09T21:12:58.000+0000
S2GRAPH-120,2016-10-10T11:44:32.000+0000
S2GRAPH-118,2016-10-09T14:38:19.000+0000
S2GRAPH-105,2016-09-01T13:36:13.000+0000
S2GRAPH-65,2016-04-18T08:55:26.000+0000
S2GRAPH-64,2016-04-18T08:43:45.000+0000
S2GRAPH-63,2016-04-18T08:36:55.000+0000
```

Porzione di csv con le informazioni riguardanti i ticket

Progettazione :Estrazione dei commit

- ❖ Per l'estrazione dei commit tramite GIT è stata utilizzata la libreria JGIT di Java
- ❖ Tramite creazione di un RevWalk è possibile creare un TreeParser che permette l'estrazione delle informazioni dal repository git

```
try( RevWalk walk = new RevWalk( git.getRepository() ) ) {  
    RevCommit commit = walk.parseCommit( commitId );  
    ObjectId treeId = commit.getTree().getId();  
    try( ObjectReader reader = git.getRepository().newObjectReader() ) {  
        return new CanonicalTreeParser( null, reader, treeId );  
    }  
}
```

- ❖ Il RevCommit è un oggetto che contiene le informazioni di un commit specifico
- ❖ Tramite i metodi del RevCommit : getShortMessage(),getCommitTime(),getId().. È possibile ricostruire le informazioni che sono necessarie alla costruzione del dataset per la creazione del Process Control Chart

Progettazione : filtrazione dei commit

- ❖ Una volta ottenuti tutti i commit con le relative informazioni questi sono stati filtrati tramite Java Regular Expressions (**REGEX**) applicato al commento del commit
- ❖ In particolare questo è stato fatto in quanto i nomi dei ticket ottenuti da JIRA presentano un pattern ben preciso : **S2GRAPH-###**
- ❖ E' stato prodotto un file JSON contenente tutti i commit del progetto S2GRAPH

```
{
  "Ticket": "No",
  "CommitName": "8c3f4a28d4681999b04575eec250d02102d3bbd6",
  "Linked": "No",
  "CommitMessage": "fix the bug: broken test",
  "NumberOfFilesTouched": 3,
  "CommitTime": "2019-01-24",
  "DiffFiles": [
    "Diff: MODIFY: s2graphql/src/test/scala/org/apache/s2graph/graphql/TestGraph.scala",
    "Diff: MODIFY: s2http/src/test/scala/org/apache/s2graph/http/AdminRouteSpec.scala",
    "Diff: MODIFY: s2http/src/test/scala/org/apache/s2graph/http/MutateRouteSpec.scala"
  ]
},
{
  "Ticket": "S2GRAPH-251",
  "CommitName": "f77a152c5bf13c4844c1b68fbe6d4d5826d8e20c",
  "Linked": "Yes",
  "CommitMessage": "[S2GRAPH-251] Provide JdbcSource/Sink",
  "NumberOfFilesTouched": 1,
  "CommitTime": "2019-01-07",
  "DiffFiles": ["Diff: MODIFY: CHANGES"]
},
{
```

Porzione di JsonObject ottenuto dall'estrazione delle informazioni

Progettazione : Mapping delle informazioni

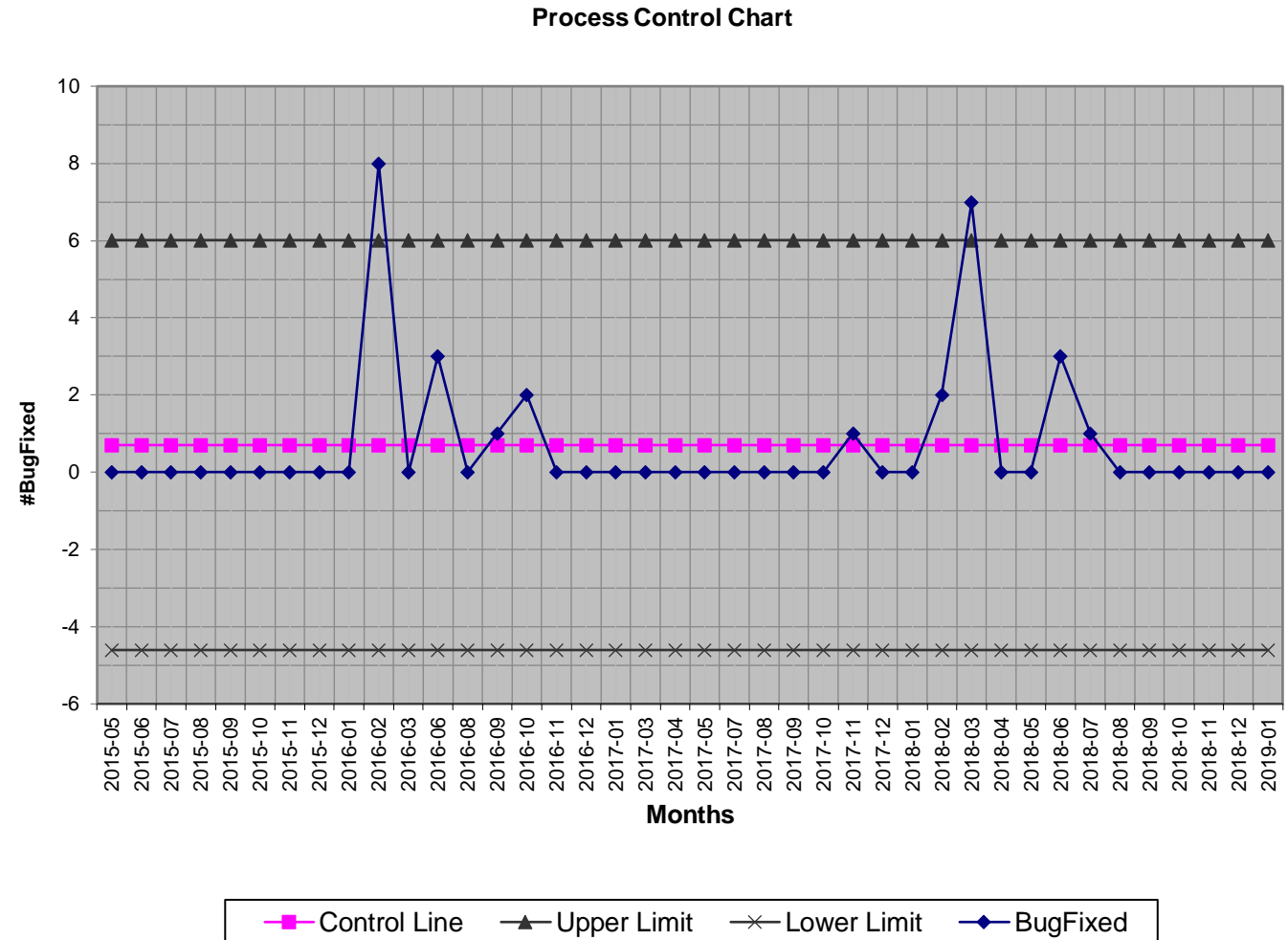
- ❖ A seguito dell'estrazione e filtraggio dei commit è stato possibile effettuare il mapping delle informazioni di JIRA e GIT
- ❖ Dai ticket con issuetype bug e status resolved sono stati selezionati solo quelli di cui è presente almeno un commit relativo
- ❖ In particolare è stato scartato il ticket S2GRAPH-171 che risulta mancante di commit
- ❖ E' stato effettuato un calcolo di linkage di qualità del dataset riportato in questa tabella:

Report qualità dataset	
Ticket mancante	S2GRAPH-171
Linkage1 = #Commit linked / #Commit	15%
Linkage2= #BugFixedTicketJira link to #TicketGit	97%
#Commit linked	248
#Commit not linked	1433
#Commit	1681

Produzione del process control chart

- ❖ Per l'asse Y (attributo) è stato utilizzata la somma del numero di commit linkati a JIRA con ticket relativo a bug fixed
- ❖ Per l'asse temporale, sono state utilizzate le date(MESE) per cui almeno 1 commit è stato effettuato nel repository. Sono state quindi escluse date mensili in cui non è presente un commit.

❖ [See details about plot](#)



Risultati

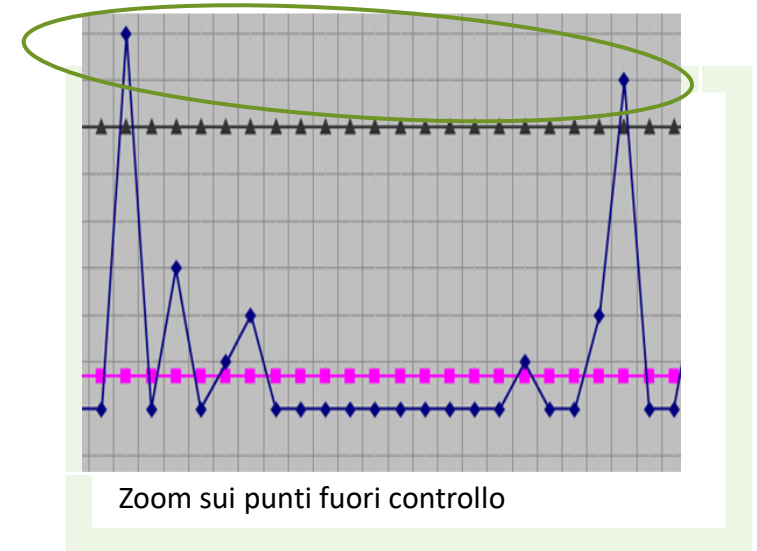
Sono stati individuati due punti fuori controllo, ovvero superiori all'UCL (limite di controllo superiore)

- Febbraio 2016 con numero di bug fixati pari a 8
- Marzo 2018 con numero di bug fixati pari a 7



Cosa indicano i punti fuori controllo?

- I due punti fuori controllo stanno ad indicare che il processo è fuori controllo – statisticamente instabile
- La presenza di un punto fuori controllo dovrebbe richiedere ulteriori indagini sul progetto
- È importante trovare le cause assegnabili e agire per rimuoverle
- Una volta rimosse le cause assegnabili dei punti fuori controllo con rimanenza delle sole variazioni all'interno dei limiti, il processo diventerà stabile e tornerà a una condizione di controllo.



Riferimenti

- Sonarcloud

[DELIVERABLE 1 ISW2 - msalvati97 \(sonarcloud.io\)](#)

- Github

[DELIVERABLE 1 ISW2/DELIVERABLE 1 ISW2 MSALVATI97 at main · msalvati97/DELIVERABLE 1 ISW2 \(github.com\)](#)