Matteo Ferretti – 0300049

**REPORT DEL PROGETTO ISW2– Deliverable 1**

Lo scopo di questo documento di testo è racchiudere i passi eseguiti per la realizzazione del progetto e le considerazioni relative che ho elaborato nel mio percorso. Procederò ora a riportare la specifica progettuale, commentare le scelte che ho eseguito per la realizzazione del codice e infine le considerazioni al riguardo con relative conclusioni.

**Indicazione:** Prelevare da Jira il progetto Falcon e sviluppare un process control chart con X = mesi e Y = numero di ticket risolti.

**Scelte progettuali (implementazione):** Ho scelto di suddividere il codice (all’interno della cartella src) in due package principali, uno contenente le classi di utility dedite a realizzare compiti specifici, l’altro contenente le classi di “controllo” dell’esecuzione del programma (dove è possibile trovare anche il main). Il codice scarica da GitHub la repository di Falcon e realizza a partire da Jyra un file csv che contiene l’ammontare dei ticket risolti associati ai diversi mesi di sviluppo. Qualora si volesse eseguire il programma su una macchina differente dalla mia ho lasciato nel Main delle stringhe che permettono di modificare il percorso relativo alla destinazione per il salvataggio dei file. Scelto il path, il parametro Booleano checkDifference serve a scegliere quale file csv realizzare (vedere il paragrafo sottostante **considerazioni**). Il Main procede con la creazione di una copia locale dei file del progetto Falcon qualora non fosse già presente (operazione utile per il controllo delle difformità tra le date di risoluzione riportate nei commit e quelle presenti su Jyra), per poi chiamare la classe IssueCounter che si occuperà di realizzare il file csv interagendo con le altre classi. La logica è quella di ottenere una matrice di interi contenente la somma dei ticket risolti associata quindi alla posizione (dato direttamente relazionabile al numero del mese). La classe responsabile di realizzare la suddetta matrice è TicketInfoGetter, la quale preleva le informazioni dei ticket presenti su Jyra e le inserisce in una matrice che ritornerà ad IssueCounter. Per ottenere le date ho estratto l’informazione interessata a partire dal commit, analizzando come esso era strutturato all’interno del file JSONstructure nella cartella csv. A seconda del parametro checkDifference poi sarà possibile scegliere se dovranno essere considerate le difformità con le date riportate nei commit (eventualmente chiamando la classe GitHandler, dedita proprio ad eseguire questo confronto con il suo metodo checkDateValidity()).

**Considerazioni**: Durante lo sviluppo del progetto ho notato che poteva accadere che le date di risoluzione di un determinato ticket potevano differire tra quelle riportate su Jyra e quelle dell’ultimo commit relativo. Per questo motivo ho introdotto all’interno del codice un parametro Booleano (modificabile nel Main) che permette di eseguire il programma in modalità confronto o in modalità non confronto. Ovviamente se si sceglie la modalità senza il confronto il tempo di esecuzione è nettamente minore, indice del fatto che deve fare una quantità di controlli esponenzialmente più piccola. Quando si imposta il parametro su true e quindi si vogliono realizzare i confronti, per ogni singolo ticket bisognerà controllare tutti i commit (operazione che rallenterà notevolmente l’esecuzione).

Una volta completata l’esecuzione del codice ho fatto in modo che venissero realizzati due file .csv all’interno della cartella apposita, proprio per tenere conto poi delle differenze fra i ticket.

Nel mio caso ci sono solo 7 difformità, ovvero:

- FALCON-86 from 8/2013 to 9/2013

- FALCON-88 from 8/2013 to 9/2013

- FALCON-98 from 3/2014 to 1/2015

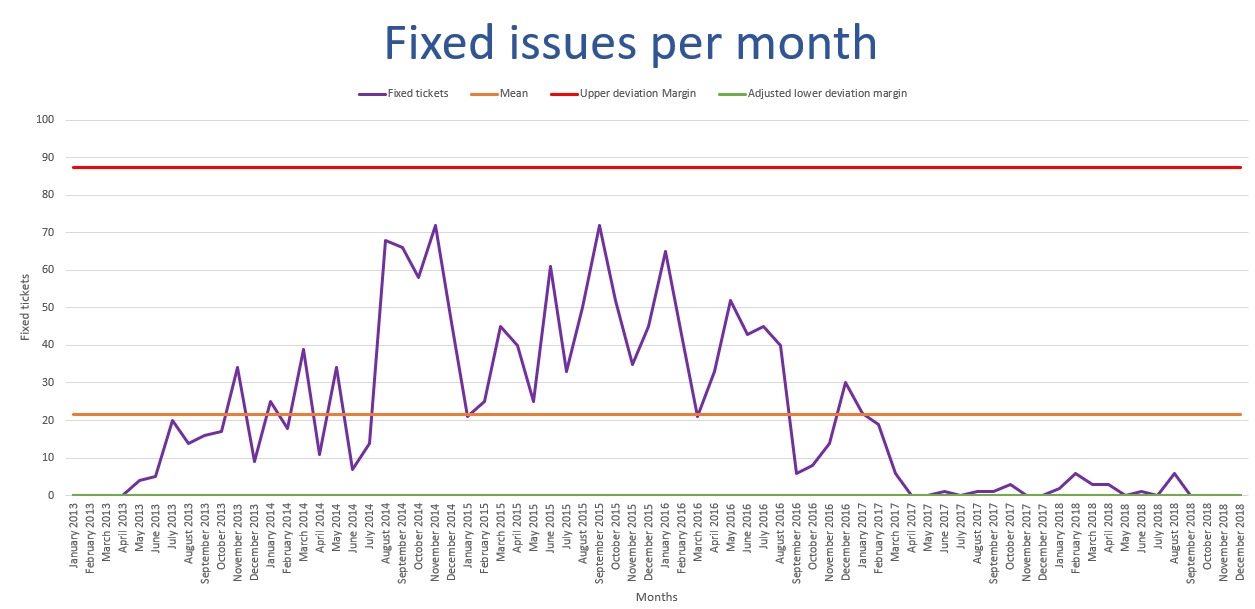
- FALCON-677 from 9/2014 to 2/2015

- FALCON-1113 from 3/2015 to 4/2015

- FALCON-2005 from 7/2016 to 8/2016

- FALCON-2280 from 2/2017 to 3/2017

Le suddette difformità sono appuntate nel file differences, nella cartella dove è possibile trovare i process control chart (ho fatto anche degli screenshot per una visione più rapida). I due grafici sono molto simili fra loro per via di questa presenza estremamente ridotta di differenze tra le date di risoluzione, quindi riporto qui soltanto quello che tiene conto delle differenze (sono entrambi ritrovabili all’interno della cartella Process Control Chart).

Il grafico riporta l’andamento dei fixed tickets per mese, insieme a media e incremento del triplo della deviazione standard, per tenere conto se in qualche mese di sviluppo qualcosa fosse mai andato particolarmente storto (ovvero ci fosse un quantitativo di fixed issues particolarmente elevato). La “deviazione standard inferiore” viene negativa per via del ridotto valore della media rispetto a quello della deviazione standard, quindi l’ho approssimata a 0.

Ho notato considerando le differenze fra i due grafici che il valore della media resta immutato (vengono effettivamente tolti ticket da mesi per essere messi in altri, di fatto non aumentando i pesi relativi al numero di ticket sul calcolo), mentre invece le deviazioni standard aumentano in modulo per via dei valori più grandi che si generano quando si considerano le differenze.

**Conclusioni**: Per una stima più appropriata del processo di sviluppo di un progetto, sarebbe più corretto considerare le difformità tra Jyra e i commit, anche se nel mio caso specifico le differenze sono minimali data la quantità ridotta di differenze. La cosa migliore è quindi avere peggiori tempi di esecuzione a fronte di stime più accurate, non trattandosi di funzionalità particolarmente critiche che richiedono ridotti tempi di calcolo e processamento.