



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Ingegneria, Gestione ed Evoluzione del Software

Test Incident Report

TEAM MEMBER

Donia Daniele - 0522501575

La Marca Antonio - 0522501557

Somma Pasquale - 0522501543

1	Introduzione	1
1.1	Approccio e Riferimenti	1
2	Risultati testing di unità	2
2.1	Repo Mining	2
2.1.1	Incident Report UTI1	2
2.1.2	Incident Report UTI2	3
2.2	Text Mining	4
2.2.1	Incident Report UTI3	4
2.2.2	Incident Report UTI4	5
2.2.3	Incident Report UTI5	7
2.2.4	Incident Report UTI6	8
2.3	Rifinitura dati analisi statica	9
2.4	Union	9
3	Risultati testing di integrazione	10
3.1	Repo Mining	10
3.1.1	Incident Report ITI1	10
3.1.2	Incident Report ITI2	11
3.1.3	Incident Report ITI3	12
3.1.4	Incident Report ITI4	13
3.2	Text Mining	14
3.2.1	Incident Report ITI4	14

3.2.2	Incident Report ITI5	16
3.2.3	Incident Report ITI6	17
3.2.4	Incident Report ITI7	18
3.3	Rifinitura dati analisi statica	20
3.4	Union	20
3.4.1	Incident Report ITI8	20
3.4.2	Incident Report ITI9	21

1.1 Approccio e Riferimenti

Nel presente report saranno dettagliati i difetti e le problematiche riscontrate nell'ambito dell'esecuzione dei casi di test. L'obiettivo principale di questo documento è quello di fornire un'analisi più dettagliata dei fault emersi durante il processo di testing preliminare a livello di unità e di integrazione. Attraverso una valutazione dettagliata dei problemi riscontrati, si mira a migliorare il processo complessivo di sviluppo. Vengono inoltre evidenziati comportamenti atipici o scelte di implementazioni non ottimali, che potrebbero essere modificate per migliorare la qualità del codice, la sua manutenibilità e l'esecuzione del tool.

Risultati testing di unità

Di seguito sono riportati gli esiti del testing di unità e le conseguenti osservazioni.

2.1 Repo Mining

2.1.1 Incident Report UTI1

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** UTI1 (Unit testing incidente 1)
- **Priorità:** Alta
- **Gravità:** Critica

2. Incidente

- **Descrizione:** eseguendo lo script `divide_dataset.py` con il file `initial_dataset.csv` non vuoto e con almeno 50 record, il sistema non riporta l'header dei dati nei CSV creati successivamente al primo.
- **Passi per riprodurre:**
 - (a) creare un file `initial_dataset.csv` con almeno 50 record.

(b) eseguire lo script `divide_dataset` da linea di comando.

- **Comportamento attuale:** l'applicazione non restituisce alcun errore.

3. Risoluzione e pianificazione

- **Stato: Risolto**
- **Cause identificate:** il problema è dovuto alla scrittura delle prime 50 linee di `initial_dataset.csv` nel primo file .csv creato, senza il salvataggio a priori dell'header del dataset e la copia nei successivi CSV.
- **Componente coinvolto:** `divide_dataset.py`
- **Pianificazione:** la modifica è minima e coinvolge una singola classe, quindi si ritiene non sia necessaria alcuna impact analysis.
- **Soluzione:**
 - aggiunta salvataggio dell'header del dataset e scrittura
 - modifica del range `for` relativo alla lunghezza del CSV (esclusione header)
 - copia header in ogni CSV creato

2.1.2 Incident Report UTI2

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** UTI2 (Unit testing incidente 2)
- **Priorità:** Bassa
- **Gravità:** Lieve

2. Incidente

- **Descrizione:** mancanza di incremento dell'indice nel file `ERRORS.txt` quando occorre un `ValueError`.
- **Passi per riprodurre:**
 - (a) Il file `1.csv` deve contenere i seguenti due record:
 - `1,https://github.com/apache/poi,d72bd78c19dfb7b57395a66ae8d9269d59a87bd2,pos`

– 2, <https://github.com/apache/santuario-java>, a09b9042f7759d094f2d49f40fc7bcf145164b25, pos

(b) Eseguire lo script `main_repo_Mining.py` da linea di comando

- **Comportamento attuale:** l'applicazione non restituisce alcun errore; tuttavia, nel file `ERRORS.txt` l'indice di linea per le due repo è lo stesso.

3. Risoluzione e pianificazione

- **Stato: Risolto**
- **Cause identificate:** l'errore è dovuto al mancato incremento della variabile `j` (corrispondente all'indice) nel `catch` responsabile della gestione dell'errore `ValueError`.
- **Componente coinvolto:** `repo_Mining.py`
- **Pianificazione:** la modifica è minima e coinvolge una singola classe, quindi si ritiene non sia necessaria alcuna impact analysis.
- **Soluzione:**
 - aggiunta dell'incremento della variabile `j` nel `catch` di `ValueError`.

2.2 Text Mining

2.2.1 Incident Report UTI3

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** UTI3 (Unit testing incidente 3)
- **Priorità:** Alta
- **Gravità:** Critica

2. Incidente

- **Descrizione:** eseguendo lo script `text_mining.py` con le directory dove sono conservate le varie repository vuote, il sistema all'interno del `for` che scandisce le varie repository cerca di accedere alla directory `RepositoryMining18` che però non esiste, interrompendo così l'esecuzione con un errore.

- **Passi per riprodurre:**
 - (a) Eseguire lo script `text_mining.py` da linea di comando.
- **Comportamento attuale:** L'applicazione restituisce un errore di tipo `FileNotFoundException`.

3. Risoluzione e pianificazione

- **Stato: Risolto**
- **Cause identificate:** Il problema è dovuto al ciclo `for` col quale il sistema cerca di accedere alle directory `RepositoryMining`, numerate da 1 a 35 esclusa per la 18esima, che però non esclude la numero 18.
- **Componente coinvolto:** `text_mining.py`
- **Pianificazione:** la modifica è minima e coinvolge una singola classe, quindi si ritiene non sia necessaria alcuna *impact analysis*.
- **Soluzione:**
 - aggiunta di un controllo sull'indice `'count'` del `for`, in modo da interrompere il flusso di esecuzione se uguale a 18, tornare al `for` e incrementare l'indice.

2.2.2 Incident Report UTI4

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** UTI4 (Unit testing incidente 4)
- **Priorità:** Media
- **Gravità:** Bassa

2. Incidente

- **Descrizione:** eseguendo lo script `text_mining.py` con tutte le directory necessarie allo script e i file java, salvati all'interno delle varie directory che rappresentano il commit da cui sono stati estratti, se è già presente un file di `text_mining.txt` lo script elaborerà sia il file java che il file di `text_mining.txt`, sovrascrivendo il file di `mining` ma creando anche un file denominato `text_mining.txt_text_mining.txt`,

contente un dizionario vuoto. Eseguendo nuovamente lo script verranno creati ulteriori file concatenando `_text_mining.txt` nel nome, contenenti sempre un dizionario vuoto.

- **Passi per riprodurre:**

- (a) Devono essere presenti tutte le directory necessarie
- (b) File java e il relativo file di text mining all'interno dell'ultima directory denominata come il codice del commit da cui sono estratti i file
- (c) Eseguire lo script `text_mining.py` da linea di comando.

- **Comportamento attuale:** L'esecuzione termina a buon fine con una corretta creazione del file di text mining, tuttavia crea un ulteriore file non previsto dovuto all'elaborazione del precedente text mining, in quanto non viene effettuato alcun controllo sul tipo di di entità contenuta nella directory analizzata e su cui verrà effettuata l'elaborazione.

3. Risoluzione e pianificazione

- **Stato: Risolto**

- **Cause identificate:** Il problema è dovuto al fatto che lo script applica il text mining a tutti i file contenuti all'interno della directory col nome del commit, con la convinzione che ci siano solamente i file java. Tuttavia, salvando i file di text mining nella stessa directory, con una successiva esecuzione dello script anche questi verranno coinvolti

- **Componente coinvolto:** `text_mining.py`

- **Pianificazione:** i file ridondanti vengono utilizzati anche dagli altri script, però applicando la modifica a `text_mining`, gli altri non ne saranno più affetti, quindi non si ritiene necessario un impact analysis.

- **Soluzione:**

- aggiunta di un controllo che permette di continuare l'elaborazione solo se il file contenuto nella directory ha estensione `.java`, in modo così da analizzare solo i file giusti e non considerare neanche possibili altre sotto-directory.

2.2.3 Incident Report UTI5

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** UTI5 (Unit testing incidente 5)
- **Priorità:** Bassa
- **Gravità:** Bassa

2. Incidente

- **Descrizione:** eseguendo lo script `text_mining.py` con tutte le directory necessarie al sistema e i file java a disposizione, il sistema quando andrà ad eseguire la funzione `removeComments` eliminerà unicamente i commenti multi-line, mentre quelli single-line verrà rimossi dalla funzione `stringTokenizer`. Ci sembra più logico accorpate nella funzione `removeComments` tutta la logica di rimozione dei commenti dal codice estratto dai file Java.
- **Passi per riprodurre:**
 - (a) Eseguire lo script `text_mining.py` da linea di comando.
- **Comportamento attuale:** L'esecuzione dello script va a buon fine, i file di `text_mining` vengono creati correttamente.

3. Risoluzione e pianificazione

- **Stato:** Risolto
- **Cause identificate:** Il problema è dovuto al fatto che la logica di rimozione dei commenti è divisa tra due funzioni, causando ridondanza e complessità nel mantenimento del codice, poiché la gestione dei commenti è frammentata e può portare a incoerenze o difficoltà nel debug e nell'estensione delle funzionalità.
- **Componente coinvolto:** `text_mining.py`
- **Pianificazione:** la modifica è minima perchè coinvolge due metodi utilizzati dallo stesso script, che vengono eseguito una dopo l'altra, quindi si ritiene non sia necessaria alcuna impact analysis.
- **Soluzione:**

- spostare la logica di eliminazione dei commenti single-line dalla funzione `stringTokenizer` a quella di `removeComments`.
- miglioramento della logica di rimozione dei commenti single-line.

2.2.4 Incident Report UTI6

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** UTI6 (Unit testing incidente 6)
- **Priorità:** Alta
- **Gravità:** Critica

2. Incidente

- **Descrizione:** eseguendo lo script `dict_generator.py` con le directory dove sono conservate le varie repository vuote, il sistema all'interno del for per scandire le varie repository cerca di accedere alla directory `RepositoryMining18` che però non esiste, interrompendo così l'esecuzione con un errore.
- **Passi per riprodurre:**
 - (a) Eseguire lo script `dict_generator.py` da linea di comando.
- **Comportamento attuale:** L'applicazione restituisce un errore di tipo `FileNotFoundException`.

3. Risoluzione e pianificazione

- **Stato:** Risolto
- **Cause identificate:** Il problema è dovuto al ciclo for col quale il sistema cerca di accedere alle directory `RepositoryMining`, numerate da 1 a 35, ad eccezione della 18esima che non esiste, che però non viene esclusa dall'iterazione.
- **Componente coinvolto:** `dict_generator.py`
- **Pianificazione:** la modifica è minima e coinvolge una singola classe, quindi si ritiene non sia necessaria alcuna impact analysis.
- **Soluzione:**
 - aggiunta di un controllo sull'indice 'count' del for, in modo da interrompere il flusso di esecuzione se uguale a 18 e tornare al for per essere incrementato.

2.3 Rifinitura dati analisi statica

Non sono state riscontrate anomalie durante l'esecuzione dei test di unità.

2.4 Union

Non sono state riscontrate anomalie durante l'esecuzione dei test di unità.

Risultati testing di integrazione

Di seguito sono riportati gli esiti del testing di integrazione e le conseguenti osservazioni.

3.1 Repo Mining

3.1.1 Incident Report ITI1

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** ITI1 (Integration testing incidente 1)
- **Priorità:** Alta
- **Gravità:** Critica

2. Incidente

- **Descrizione:** L'esecuzione dello script `main_repo_Mining.py` avendo il file `initial_dataset.csv` non vuoto contenente 0 record, si ottiene un `FileNotFoundError`.
- **Passi per riprodurre:**
 - (a) Creare il file `initial_dataset.csv` con 0 record
 - (b) Eseguire lo script `divide_dataset.py` da linea di comando

(c) Eseguire lo script `main_repo_Mining.py` da linea di comando

- **Comportamento attuale:** L'applicazione restituisce un `FileNotFoundError`.

3. Risoluzione e pianificazione

- **Stato: Risolto**
- **Cause identificate:** In `main_repo_Mining.py` il numero di iterazioni è fisso e non dipende dal numero delle repository ottenute eseguendo `divide_dataset.py`. Quindi, con un numero di repository pari a 0, `initialize()` è chiamato ugualmente, creando la directory 'RepositoryMining1'. Vi è quindi una seguente iterazione che causa il `FileNotFoundError` poiché il sistema non trova la directory 'Dataset_Divided'.
- **Componente coinvolto:** `main_repo_Mining.py`, `repo_Mining.py`
- **Pianificazione:** La modifica è minima e coinvolge una singola classe, quindi si ritiene non sia necessaria alcuna impact analysis.
- **Soluzione:**
 - Modifica del range nel ciclo all'interno di `mining_result` con calcolo del numero di repository esistenti.

3.1.2 Incident Report ITI2

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** ITI2 (Integration testing incidente 2)
- **Priorità:** Alta
- **Gravità:** Critica

2. Incidente

- **Descrizione:** Eseguendo lo script `main_repo_Mining.py` con il file `initial_dataset.csv` non vuoto e con più di 50 record, alla seconda iterazione la funzione `initialize()` restituisce un `FileNotFoundError`, in quanto non viene trovata la directory `Dataset_Divided`.

- **Passi per riprodurre:**
 - (a) Creare un file `initial_dataset.csv` con almeno 50 record
 - (b) Eseguire lo script `divide_dataset.py` da linea di comando
 - (c) Eseguire lo script `main_repo_Mining.py` da linea di comando
- **Comportamento attuale:** L'applicazione restituisce un `FileNotFoundError`.

3. Risoluzione e pianificazione

- **Stato: In corso**
- **Cause identificate:** Il problema è dovuto alla navigazione del file system definita nella funzione `initialize()`, in particolare l'istruzione `os.chdir('..')` che, alla seconda iterazione, provoca lo spostamento nella directory superiore a quella in cui si trova il file ricercato.
- **Componente coinvolto:** `main_repo_mining.py`, `repo_Mining.py`
- **Pianificazione:** La modifica coinvolge esclusivamente lo script `repo_Mining.py`, quindi non si ritiene necessaria l'impact analysis.
- **Soluzione:**
 - Modifica logica in `initialize` per settare e ripristinare directory corrente prima e dopo l'invocazione di `startMiningRepo`

3.1.3 Incident Report ITI3

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** ITI3 (Integration testing incidente 3)
- **Priorità:** Alta
- **Gravità:** Critica

2. Incidente

- **Descrizione:** Eseguendo lo script `main_repo_Mining.py` con il file `initial_dataset.csv` non vuoto, contenente il link a una repository con commit contenente modifiche a file java, si ottiene un `FileNotFoundError`.

- **Passi per riprodurre:**
 - (a) Creare il file `initial_dataset.csv` con il seguente record:
 - 1, link repo: `https://github.com/winterbe/java8-tutorial`,
`81a0fa3aa1d6ec2409e0226d3a6c2f5c2d19a41d`, pos
 - (b) Eseguire lo script `divide_dataset.py` da linea di comando
 - (c) Eseguire lo script `main_repo_Mining.py` da linea di comando
- **Comportamento attuale:** L'applicazione restituisce un `FileNotFoundError`.

3. Risoluzione e pianificazione

- **Stato:** In corso
- **Cause identificate:** Quando sono estratte classi con estensione `.java`, il sistema tenta di spostarsi nella directory in cui salvare le classi, utilizzando il parametro `cwd`. Il percorso specificato da `cwd` non è esistente, in quanto non include la directory `mining_results`, generando `FileNotFoundError`.
- **Componente coinvolto:** `repo_mining.py`
- **Pianificazione:** La modifica coinvolge il metodo `startMiningRepo` e riguarda la navigazione all'interno file system e la manipolazione dei parametri in input. Si ritiene quindi necessaria l'impact analysis.
- **Soluzione:**
 - estrazione dal parametro `cwd` del percorso padre
 - specifica directory corrente come concatenazione del percorso estratto con `mining_results`

3.1.4 Incident Report ITI4

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** ITI4 (Integration testing incidente 4)
- **Priorità:** Media
- **Gravità:** Media

2. Incidente

- **Descrizione:** L'esecuzione dello script `main_repo_Mining.py` con la directory `mining_results` già esistente e non vuota, comporta la presenza nei risultati del mining di repository aggiuntive rispetto a quelle presenti in `initial_dataset`.
- **Passi per riprodurre:**
 - (a) Creare il file `initial_dataset.csv` con 50 record
 - (b) Creare la directory `mining_results` non vuota
 - (c) Eseguire lo script `divide_dataset.py` da linea di comando
 - (d) Eseguire lo script `main_repo_Mining.py` da linea di comando
- **Comportamento attuale:** L'applicazione non restituisce alcun errore, ma i risultati del mining non corrispondono a quelli attesi.

3. Risoluzione e pianificazione

- **Stato: Risolto**
- **Cause identificate:** In `repo_Mining.py`, in particolare nella funzione `initialize`, prima dell'inizio del mining si entra nella directory `mining_results` senza verificare se sono presenti risultati di precedenti estrazioni.
- **Componente coinvolto:** `repo_Mining.py`
- **Pianificazione:** La modifica è minima e coinvolge una singola classe, quindi si ritiene non sia necessaria alcuna impact analysis.
- **Soluzione:**
 - Aggiunta eliminazione e successiva creazione di `mining_results` se presente.

3.2 Text Mining

3.2.1 Incident Report ITI4

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** ITI4 (Integration testing incidente 4)
- **Priorità:** Media
- **Gravità:** Critica

2. Incidente

- **Descrizione:** eseguendo tutti gli script del modulo di Text_Mining in ordine, con tutte le directory necessarie agli script e i file java, salvati all'interno delle varie directory che rappresentano il commit da cui sono stati estratti, viene effettuato il text mining per tutte le directory tranne per RepositoryMining1.
- **Passi per riprodurre:**
 - (a) Devono essere presenti tutte le directory necessarie
 - (b) File java e il relativo file di text mining all'interno dell'ultima directory denominata come il codice del commit da cui sono estratti i file
 - (c) Eseguire lo script `text_mining.py` da linea di comando.
 - (d) Eseguire lo script `dict_generator.py` da linea di comando.
 - (e) Eseguire lo script `less_element_text_mining.py` da linea di comando.
 - (f) Eseguire lo script `creator_csv_for_TextMining.py` da linea di comando.
- **Comportamento attuale:** L'applicazione viene eseguita correttamente senza errori, ma i file di text mining (già presenti da qualche precedente esecuzione) dentro RepositoryMining1 non vengono aggiornati, in quanto non considerato nello script di `text_mining.py`.

3. Risoluzione e pianificazione

- **Stato: Risolto**
- **Cause identificate:** Il problema è dovuto al fatto che nel for più esterno durante lo spostamento del sistema nelle varie directory, l'indice 'count' varia nell'intervallo da 2 a 35.
- **Componente coinvolto:** `text_mining.py`
- **Pianificazione:** la modifica coinvolge solamente il metodo main dello script `text_mining.py`, in quanto è l'unico a non considerare la prima repository. Gli altri script considerano RepositoryMining1 nella loro elaborazione e non incappano in errori perchè utilizzano file di text_mining precedentemente ottenuti. Per quest ragione non si ritiene opportuno effettuare l'impact analysis.
- **Soluzione:**
 - modifica dell'intervallo tra cui varia l'indice 'count', facendolo partire da 1.

3.2.2 Incident Report ITI5

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** ITI5 (Integration testing incidente 5)
- **Priorità:** Alta
- **Gravità:** Critica

2. Incidente

- **Descrizione:** eseguendo tutti gli script del modulo di Text_Mining in ordine, con tutte le directory necessarie agli script e i file java, salvati all'interno delle varie directory che rappresentano il commit da cui sono stati estratti, quando il sistema proverà ad accedere ad un file sia in lettura che in scrittura, fallirà.
- **Passi per riprodurre:**
 - (a) Devono essere presenti tutte le directory necessarie
 - (b) File java e il relativo file di text mining all'interno dell'ultima directory denominata come il codice del commit da cui sono estratti i file
 - (c) Eseguire lo script `text_mining.py` da linea di comando.
 - (d) Eseguire lo script `dict_generator.py` da linea di comando.
 - (e) Eseguire lo script `less_element_text_mining.py` da linea di comando.
 - (f) Eseguire lo script `creator_csv_for_TextMining.py` da linea di comando.
- **Comportamento attuale:** L'applicazione restituisce un errore di tipo `UnicodeDecodeError`, interrompendo l'esecuzione.

3. Risoluzione e pianificazione

- **Stato:** Risolto
- **Cause identificate:** Il problema è dovuto al fatto che nella lettura o scrittura dei file, nessuno script specifica la modalità di encoding, necessaria visto che nel codice dei file java sono presenti caratteri speciali.
- **Componente coinvolto:** `text_mining.py`, `dict_generator.py`, `less_element_text_mining.py` e `creator_csv_for_TextMining.py`.

- **Pianificazione:** dal momento che tutti gli script leggono e scrivono su file, la modifica li coinvolge tutti. Necessita quindi di un impact analysis.
- **Soluzione:**
 - aggiunta, in modo esplicito, della modalità di encoding 'utf-8' all'interno di tutte le funzioni di open presenti negli script di questo modulo.

3.2.3 Incident Report ITI6

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** ITI6 (Unit testing incidente 6)
- **Priorità:** Media
- **Gravità:** Media

2. Incidente

- **Descrizione:** eseguendo tutti gli script del modulo di Text_Mining in ordine, con tutte le directory necessarie agli script, i file java e già il relativo file di text_mining.txt, verrà creato (come spiegato nell'UTI4), un ulteriore file dall'esecuzione dello script text_mining.py, denominato text_mining.text_text_mining.txt e quest'ultimo avrà una propria riga all'interno del csv finale creato dallo script "creator_csv_for_TextMining.py, con 0 in ogni colonna.
- **Passi per riprodurre:**
 - (a) Devono essere presenti tutte le directory necessarie
 - (b) File java e il relativo file di text mining all'interno dell'ultima directory denominata come il codice del commit da cui sono estratti i file
 - (c) Eseguire lo script `text_mining.py` da linea di comando.
 - (d) Eseguire lo script `dict_generator.py` da linea di comando.
 - (e) Eseguire lo script `less_element_text_mining.py` da linea di comando.
 - (f) Eseguire lo script `creator_csv_for_TextMining.py` da linea di comando.

- **Comportamento attuale:** L'esecuzione termina a buon fine con una corretta creazione del file di text mining, tuttavia crea un ulteriore file non previsto e quest'ultimo è presente anche all'interno del csv finale. Molte esecuzioni degli script portano ad una crescita spropositata del csv finale, con righe del tutto inutili, inficiando potenzialmente sulla memoria utilizzata e i tempi di elaborazione.

3. Risoluzione e pianificazione

- **Stato: Risolto**
- **Cause identificate:** Il problema è dovuto al fatto che lo script finale crea il csv sulla base di tutti i file che contengono la stringa "text_mining.txt" nel nome.
- **Componente coinvolto:** `text_mining.py` e `creator_csv_for_TextMining.py`
- **Pianificazione:** la problematica è legata a quella individuata nell'UTI3, quindi risolvendo quella anche questa di conseguenza viene risolta, per questo non si ritiene necessario un impact analysis. Per precauzione verrà applicata una modifica ad entrambe le classi main dei due script sopracitati, così da affinare i controlli.
- **Soluzione:**
 - aggiunta di un controllo in `text_mining.py` che permette di continuare l'elaborazione solo se il file contenuto nella directory ha estensione .java, in modo così da analizzare solo i file giusti e non considerare neanche possibili altre sotto-directory.
 - modifica del controllo in `creator_csv_for_TextMining.py`, così da fargli considerare nella creazione del csv solamente i file che terminano per ".java_text_mining.txt".

3.2.4 Incident Report ITI7

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** ITI7 (Integration testing incidente 7)
- **Priorità:** Bassa
- **Gravità:** Bassa

2. Incidente

- **Descrizione:** eseguendo tutti gli script del modulo di Text_Mining in ordine, con tutte le directory necessarie allo script e i file java, salvati all'interno delle varie directory che rappresentano il commit da cui sono stati estratti, alla creazione del csv, nella colonna relativa al nome del file ritroviamo appiunto il nome del file seguito per da un '_'.
- **Passi per riprodurre:**
 - (a) Devono essere presenti tutte le directory necessarie.
 - (b) File java all'interno dell'ultima directory denominata come il codice del commit da cui sono estratti i file.
 - (c) Eseguire lo script `text_mining.py` da linea di comando.
 - (d) Eseguire lo script `dict_generator.py` da linea di comando.
 - (e) Eseguire lo script `less_element_text_mining.py` da linea di comando.
 - (f) Eseguire lo script `creator_csv_for_TextMining.py` da linea di comando.
- **Comportamento attuale:** L'applicazione viene eseguita correttamente senza errori, vengono creati tutti i file risultanti dall'elaborazione degli script, ma nel csv finale creato nella colonna relativa al nome del file ritroviamo appunto il nome del file seguito per da un '_'. Che si presenterà anche nei csv creati dagli script del modulo Union, che utilizzano il csv creato dal Text Mining.

3. Risoluzione e pianificazione

- **Stato: Risolto**
- **Cause identificate:** Il problema è dovuto che durante la creazione del csv, il nome del file si ottiene dal nome del file di text mining eliminando la parte finale del nome, ovvero 'text_mining.txt', dimenticandosi però del '_' precedente.
- **Componente coinvolto:** `creator_csv_for_TextMining.py`, `Union_TMwithASA.py`, `Union_TMwithSM.py` e `Total_Combination.py`
- **Pianificazione:** la modifica coinvolge principalmente lo script di creazione del csv, che è l'origine del problema, quindi non necessita di un impact analysis. Gli script di union, presentano del codice per eliminare '_', che diventerà inutile in seguito alla modifica.

- **Soluzione:**
 - il nome all'interno del file csv viene ricavato dal nome del file di text mining, quindi basta cambiare l'istruzione `file.replace("text_mining.txt", "")` in `file.replace("_text_mining.txt", "")`.
 - eliminare l'istruzione `line_tm.split(',')[0].replace("_", "")`, che non causa errori, ma che diventa inutile.

3.3 Rifinitura dati analisi statica

Non sono state riscontrate anomalie durante l'esecuzione dei test di integrazione.

3.4 Union

3.4.1 Incident Report ITI8

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** ITI8 (Integration testing incidente 8)
- **Priorità:** Alta
- **Gravità:** Critica

2. Incidente

- **Descrizione:** eseguendo lo script `Union.py`, con i file derivanti dal modulo Text Mining e dal modulo Software Metrics, quando il sistema proverà ad accedere in lettura al file relativo al Text Mining, fallirà.
- **Passi per riprodurre:**
 - (a) Devono essere presenti tutte le directory necessarie.
 - (b) File csv, derivante dal modulo Text Mining, e il relativo file csv, inerente alle Software Metrics.
 - (c) Eseguire lo script `Union.py` da linea di comando.
- **Comportamento attuale:** L'applicazione restituisce un errore di tipo `UnicodeDecodeError`, interrompendo l'esecuzione.

3. Risoluzione e pianificazione

- **Stato: Risolto**
- **Cause identificate:** Il problema è dovuto al fatto che nella lettura del file csv inerente al Text Mining non viene specificata la modalità di encoding, necessaria visto che sono presenti caratteri speciali.
- **Componente coinvolto:** `Union.py`
- **Pianificazione:** dal momento che tutti gli script leggono e scrivono su file, la modifica li coinvolge tutti. Necessita quindi di un impact analysis.
- **Soluzione:**
 - aggiunta, in modo esplicito, della modalità di encoding 'utf-8' all'interno di tutte le funzioni di open presenti negli script di questo modulo.

3.4.2 Incident Report ITI9

1. Informazioni di base

- **Progetto:** 'Perseverance'
- **Ambiente:** Ambiente di test
- **Nome incidente:** ITI9 (Integration testing incidente 9)
- **Priorità:** Alta
- **Gravità:** Critica

2. Incidente

- **Descrizione:** eseguendo lo script `Union_TMwithASA.py`, con i file derivanti dal modulo Text Mining e dal modulo `mining_results_asa`, quando il sistema proverà ad accedere in lettura al file relativo al Text Mining, fallirà.
- **Passi per riprodurre:**
 - (a) Devono essere presenti tutte le directory necessarie.
 - (b) File csv, derivante dal modulo Text Mining, e il relativo file csv derivante da `mining_results_asa`
 - (c) Eseguire lo script `Union_TMwithASA.py` da linea di comando.
- **Comportamento attuale:** L'applicazione restituisce un errore di tipo `UnicodeDecodeError`, interrompendo l'esecuzione.

3. Risoluzione e pianificazione

- **Stato: Risolto**
- **Cause identificate:** Il problema è dovuto al fatto che nella lettura del file csv inerente al Text Mining non viene specificata la modalità di encoding, necessaria visto che sono presenti caratteri speciali.
- **Componente coinvolto:** `Union_TMwithASA.py`
- **Pianificazione:** dal momento che tutti gli script leggono e scrivono su file, la modifica li coinvolge tutti. Necessita quindi di un impact analysis.
- **Soluzione:**
 - aggiunta, in modo esplicito, della modalità di encoding 'utf-8' all'interno di tutte le funzioni di open presenti negli script di questo modulo.