



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof.ssa F.Ferrucci



TSR Test Summary Report

Riferimento	
Versione	1.0
Data	3/02/2020
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Cirillo Franco Cirillo Luigi Fusco Ciro Aiello Vincenzo
Approvato da	



Revision History

Data	Versione	Cambiamenti	Autori
17/01/2020	1.0	Prima stesura	Cirillo Franco



Sommario

I.	1. Introduzione	4
	1.1 Glossario	4
	1.2 Acronimi ed Abbreviazioni	4
	1.3 Panoramica.....	4
II.	2. Relazione con altri documenti di testing.....	5
	2.1 Relazioni con il Test Plan (TP).....	5
	2.2 Relazioni con il Test Case Specification (TCS)	5
	2.3 Test Execution Report (TER)	5
	2.4 Relazioni con il Test Incident Report (TIR).....	5
	2.5 Relazioni con il Test Case Integration Document (TCID)	5
III.	3. Analisi del testing di unità	5
IV.	4. Analisi del testing di integrazione	6
V.	5. Analisi dei Test Case.....	7
VI.	6. Passed test cases	8
VII.	7. Analisi del testing di usabilità.....	8

1. Introduzione

Nel seguente documento viene mostrato il resoconto delle attività di testing ed i relativi risultati finali fornendo anche una valutazione relativamente all'esecuzione dei test case specificati nel documento Test Case Specification. Effettuiamo dunque, una prima analisi dei difetti riscontrati nell'applicazione, ovvero un insieme di valutazioni rispetto ai fallimenti o gli errori riscontrati nei test, che necessitano di ulteriori approfondimenti.

È necessario fornire un report significativo delle attività di testing svolte in precedenza in modo da poter mettere a conoscenza chi usufruirà di NewDM di eventuali failure o bug nel sistema.

I risultati dei test di sistema effettuati attraverso l'utilizzo del tool Katalon Studio, sono riportati all'interno del documento Test Execution Report. Nel seguente documento, ci serviamo di tali esiti positivi o negativi per poter giungere a soluzioni ai limiti riscontrati nei test.

1.1 Glossario

- **Test Plan:** Un Test Plan è un documento che descrive gli obiettivi, le risorse e i processi per un test specifico per un prodotto software o hardware.
- **Test Case Integration Document:** Il Test Case Integration Document riassume come verrà effettuato il testing di integrazione
- **Test Case Specification:** Il Test Case Specification riassume quale scenario sarà testato, come verranno effettuati i test e quanto spesso.
- **Test Execution Report:** Il Test Execution Report mostra l'output dei test case definiti nel documento Test Case Specification, evidenziando la differenza tra il comportamento previsto nell'oracolo e quello osservato durante l'esecuzione del test.
- **Test Incident Report:** Il Test Incident Report documenta tutte le problematiche trovate durante questa fase di test. Questo documento specifica i risultati previsti dal test, quando un test fallisce e qualsiasi indicazione del perché un test fallisce.
- **Test Summary Report:** Il Test Summary Report è il documento che contiene il resoconto delle attività di testing ed i relativi risultati finali fornendo una valutazione relativamente all'esecuzione dei test.

1.2 Acronimi ed Abbreviazioni

- **TP:** Test Plan
- **TCID:** Test Case Integration Document
- **TCS:** Test Case Specification
- **TER:** Test Execution Report
- **TIR:** Test Incident Report
- **TSR:** Test Summary Report

1.3 Panoramica

Nei successivi capitoli viene mostrato come il seguente documento si relaziona con gli altri documenti prodotti durante le attività di testing specificando come il Test Summary Report ne utilizza i risultati forniti.



2. Relazione con altri documenti di testing

In questa sezione vengono riportate le relazioni che il seguente documento ed altri documenti prodotti durante la fase di testing.

2.1 Relazioni con il Test Plan (TP)

Il Test Plan verrà utilizzato per ricavare le componenti e le funzionalità che devono essere testate.

2.2 Relazioni con il Test Case Specification (TCS)

Il Test Case Specification verrà utilizzato per ricavare le specifiche dei test cases che saranno usate per testare le funzionalità del sistema.

2.3 Test Execution Report (TER)

Il TER verrà invece utilizzato per scoprire qual è stato l'esito dei test cases delle funzionalità testate, e controllare se il risultato ottenuto corrisponde con quello desiderato.

2.4 Relazioni con il Test Incident Report (TIR)

Nel Test Incident Report saranno riportate tutte le anomalie riscontrate durante il testing delle funzionalità effettuato in questa fase.

2.5 Relazioni con il Test Case Integration Document (TCID)

Test Case Integration Document: Il Test Case Integration Document riassume come verrà effettuato il testing di integrazione

3. Analisi del testing di unità

Le nostre componenti sono testate secondo il metodo white-box, in particolare abbiamo utilizzato il metodo branch coverage e il plugin Jacoco per calcolarne il totale. I risultati ottenuti sono stati: una copertura dei branch di tutto il sistema del 100%, escludendo alcune componenti off-the-shelf.

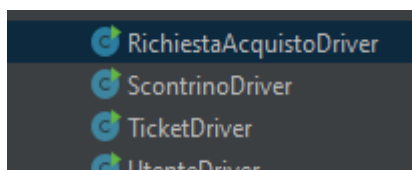
NewDMApp

Element	Missed Instructions	Cov.	Missed Branches	Cov.
business_inventario		68%		100%
business_assistenza		73%		100%
persistenza_dao		82%		100%
business_utenza		43%		100%
business_cassa		80%		100%
presentazione		33%		n/a
business_fornitura		54%		n/a
persistenza		83%		100%
exceptions		100%		n/a
Total	982 of 3.576	72%	0 of 160	100%

4. Analisi del testing di integrazione

Per effettuare l'integration testing abbiamo utilizzato la strategia bottom-up: ciò ci ha permesso di garantire la presenza di fondamenta solide alla base del sistema ma richiede di mettere in campo test driver per simulare le componenti dei layer più in alto che non sono stati ancora integrati.

Dopo aver utilizzato i driver, essi sono stati eliminati dal codice perché non più utili per lo sviluppo del progetto. Ne riportiamo alcuni esempi nelle seguenti schermate (alcuni driver e uno snippet di codice di uno di essi):



```
do {
    System.out.println("1 : Crea una nuova richiesta di acquisto");
    System.out.println("2 : Salva la richiesta");
    System.out.println("-1 : Esci");
    i = in.nextInt();
    in.nextLine();
    switch (i) {
        case 1:
            r = new RichiestaAcquisto();
            System.out.println("Inserire codice prodotto");
            r.setCodice(in.nextLong());
            System.out.println("Inserire quantità da acquistare");
            r.setQuantity(in.nextInt());
            System.out.println(r);
            break;

        case 2:
            if (r == null) {
                System.out.println("Crea prima la richiesta");
            } else {
                r.save();
                System.out.println("Richiesta salvata!");
            }
            break;
    }
}
```

5. Analisi dei Test Case

All'interno del documento Test Plan, attraverso la tecnica del category partition, sono state definite le varie combinazioni per i possibili input all'interno del sistema. Successivamente, nel documento Test Case Specification sono stati specificati in dettaglio i vari test case con il relativo comportamento atteso.

Avevamo pianificato esattamente 40 casi di test ed il 100% di questi sono stati implementati. Allo stesso modo, abbiamo eseguito tutti i test pianificati ed implementati.

Ricordiamo che con Failed sono stati indicati quei test che forniscono un risultato che coincide con il comportamento atteso specificato nell'oracolo, viceversa con Passed sono stati indicati i test che presentano un output differente da quello atteso.

I test case che hanno fornito come esito Failed sono stati 39 (cioè il quindi il 97,5% dei casi di test eseguiti). Soltanto il 2,5% dei casi di test ha osservato un comportamento differente da quello atteso, ovvero un solo test case ha fornito esito Passed.

Una volta effettuata la correzione del fault che provocava il fallimento, sono stati rieseguiti tutti i test, ottenendo così un risultato positivo al 100%.

TEST SUITE: TestSuiteAssistenza

Full Name: TestSuiteAssistenza
Start / End / Elapsed: 2021-02-01 13:09:49.533 / 2021-02-01 13:29:08.229 / 00:19:18.696
Status: 16 test total, 16 passed, 0 failed, 0 error, 0 incomplete, 0 skipped

TEST SUITE: TestSuiteCassa

Full Name: TestSuiteCassa
Start / End / Elapsed: 2021-02-01 13:30:06.566 / 2021-02-01 13:32:12.393 / 00:02:05.827
Status: 3 test total, 3 passed, 0 failed, 0 error, 0 incomplete, 0 skipped

TEST SUITE: TestSuiteMagazzino

Full Name: TestSuiteMagazzino
Start / End / Elapsed: 2021-02-01 13:33:58.470 / 2021-02-01 13:42:19.523 / 00:08:21.053
Status: 9 test total, 9 passed, 0 failed, 0 error, 0 incomplete, 0 skipped

TEST SUITE: TestSuiteLogin

Full Name: TestSuiteLogin
Start / End / Elapsed: 2021-02-02 20:17:39.532 / 2021-02-02 20:19:31.651 / 00:01:52.119
Status: 3 test total, 3 passed, 0 failed, 0 error, 0 incomplete, 0 skipped



TEST SUITE: TestSuiteInsProdottoCassa

Full Name: TestSuiteInsProdottoCassa
Start / End / Elapsed: 2021-02-02 20:20:03.607 / 2021-02-02 20:21:44.590 / 00:01:40.983
Status: 3 test total, 3 passed, 0 failed, 0 error, 0 incomplete, 0 skipped

TEST SUITE: TestSuiteAggiornaQuantitaProdotto

Full Name: TestSuiteAggiornaQuantitaProdotto
Start / End / Elapsed: 2021-02-02 20:22:05.081 / 2021-02-02 20:25:21.953 / 00:03:16.872
Status: 6 test total, 6 passed, 0 failed, 0 error, 0 incomplete, 0 skipped

6. Passed test cases

Come detto in precedenza, soltanto un caso di test non ha tenuto il comportamento specificato nell'oracolo. Il test case in questione è TC_A_1_4.

Abbiamo testato la funzionalità del sistema che permette la creazione di un ticket da parte del magazziniere inserendo un input sbagliato sul numero di telefono.

La creazione del ticket non può avere successo in quanto nel numero di telefono sono presenti lettere e quindi non rispetta il formato.

Ma quale era il comportamento atteso? Il Sistema visualizza un messaggio di errore: "Inserire un numero di telefono valido".

Cosa succedeva invece? Il Sistema visualizza un messaggio di errore: "Inserire un nome prodotto valido". Indicando quindi il corretto passaggio all'istruzione successiva.

Una volta effettuata la correzione del fault che provocava il fallimento, è stato effettuato un ulteriore test con gli stessi valori di input: questo ha dato esito positivo, cioè il sistema si è comportato secondo le aspettative.

7. Analisi del testing di usabilità

È stata adottata una tecnica che consiste nel far utilizzare il sistema ad un determinato numero di persone, assegnandogli dei task, e poi gli è stato somministrato un sondaggio per ricavare il grado di facilità nell'utilizzo del sistema, utilizzando i moduli di Google.

Il sondaggio è stato proposto ad un quantitativo di sole 10 persone, dato il ridotto budget di ore a nostra disposizione. La risposta ad ogni domanda è un valore da 1 a 5, dove 1 corrisponde ad una scarsa facilità di utilizzo e 5 ad un facile utilizzo. È stata poi fatta una media dei risultati per ogni sezione di lavoro.



Il risultati sono i seguenti:

- Cassa: 4.9/5
- Magazzino: 4.7/5
- Ticket: 4.5/5
- Rifornimento: 4.5/5

Di seguito viene riportato una sezione del modulo Google per il sondaggio

Test Uso NewDM

Entra nella sezione Magazzino

Nella sezione Magazzino prova ad aggiornare la quantità e il prezzo di un prodotto usando questo codice:
1000000000001.
Dopo di che, creane uno nuovo.

Quanto ti è sembrato facile aggiornare la quantità del prodotto?

	1	2	3	4	5	
Difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Facile

Quanto ti è sembrato facile aggiornare il prezzo del prodotto?

	1	2	3	4	5	
Difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Facile

Quanto ti è sembrato facile aggiungere un nuovo prodotto?

	1	2	3	4	5	
Difficile	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Facile

IndietroAvanti