



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica

TEST DESIGN DOCUMENT

INGEGNERIA, GESTIONE ED EVOLUZIONE DEL SOFTWARE

DOCENTI

Prof. Andrea De Lucia

TUTOR

Dott. Stefano Lambiase

Università degli Studi di Salerno

STUDENTI

Francesco Maria Torino

(0522501879)

Francesco Alessandro

Pinto (0522501981)

Stefano Guida

(0522502054)

Indice

Elenco delle Figure	ii
Elenco delle Tabelle	iii
1 Introduzione	1
1.1 Introduzione al Test Design	1
2 Test Cases	3
2.1 Test Case: Get Request [RF5]	3
3 Test Suite	7
3.1 Funzionalità: Get Request [RF5]	7
Bibliografia	11

Elenco delle figure

Elenco delle tabelle

2.1	Category Partitioning: interpretazione di una richiesta get request . .	3
3.1	Risultati dei test per le richieste <code>get_request</code>	7

CAPITOLO 1

Introduzione

1.1 Introduzione al Test Design

La seguente sezione del documento di progettazione dei test è dedicata alla definizione e specifica dei test relativi al requisito funzionale **RF5: Get Request** del sistema **GUiDO**, integrato con il tool **Community Inspector (TOAD)**. In particolare, viene adottata la tecnica del **Category Partitioning**, una metodologia strutturata di progettazione dei casi di test, che consente di suddividere in modo sistematico il dominio di input in categorie significative e scelte rilevanti, al fine di individuare combinazioni di input rappresentative e significative per la validazione del comportamento del sistema.

Nel dettaglio, il requisito funzionale **RF5** prevede che l'utente possa:

- Avviare un'analisi su una repository GitHub specificando una finestra temporale di tre mesi, attraverso il modulo *Community Inspector*;
- Ottenere, al termine dell'analisi, un *report dettagliato* contenente i *Community Smells* rilevati, le *metriche quantitative* (ad esempio dispersione, engagement, longevità, ecc.), e la *visualizzazione grafica* della rete sociale dei membri della community coinvolta nel progetto.

La progettazione dei test è articolata in conformità ai seguenti sottocapitoli:

- **Test Case: Get Request [RF5]:** viene introdotta la tecnica del *Category Partitioning* applicata al requisito, individuando i parametri rilevanti e le rispettive categorie di valori (validi e non validi).
- **Category Partitioning: interpretazione di una richiesta get request:** la tabella esplicita le *categorie individuate* per ciascun parametro in ingresso, e le relative *scelte* (comprese condizioni errate), fornendo così una visione completa del dominio da testare.
- **Test Frame:** vengono derivati i *Test Frame*, ovvero le combinazioni significative di valori delle categorie, che fungono da base per la generazione dei casi di test.

Tale approccio garantisce una copertura sistematica ed esaustiva delle possibili condizioni operative, contribuendo in modo rigoroso alla validazione del comportamento del sistema in conformità ai requisiti previsti.

Test Cases

2.1 Test Case: Get Request [RF5]**Tabella 2.1:** Category Partitioning: interpretazione di una richiesta get request

Descrizione	
Il sistema deve interpretare correttamente una richiesta di tipo <code>get_request</code>	
Parametro: URL Repository	
Nome Categoria	Scelte
Repository URL without Author or Repository Name [RA]	1. URL = <code>/^https:\\/\\/github\\.com\\ /[^/]+\\+\\/?\$/</code> [ERROR] 2. URL = <code>/^https:\\/\\/github\\.com\\ /[^/]+[^/]+\\+\\/?\$/</code> [RA_OK]
Nome Categoria	Scelte

Repository URL senza https://github.com/ [RH]	1. URL = (?!/^\https:\\/\\/github\\.com\\/) [^/]+\+\\/?\$/ [ERROR] 2. URL = /^https:\\/\\/github\\.com\\/[^/]+[^/]+\+\\/?\$/ [RH_OK]
Nome Categoria	Scelte
Repository URL Empty [RE]	1. URL = ϵ [ERROR] 2. URL $\neq \epsilon$ [RE_OK]
Parametro: Data Fine	
Nome Categoria	Scelte
Data Fine [DF]	1. DF > TODAY [ERROR] 2. DF = ϵ [ERROR] 3. DF \leq TODAY [DF_OK]
Parametro: Repository	
Nome Categoria	Scelte
Commit [CO]	1. CO < 100 [ERROR] 2. CO \geq 100 [CO_OK] 3. CO non elaborato
Parametro: Repository	
Nome Categoria	Scelte
Milestones [MI]	1. MI = 0 [ERROR] 2. MI \geq 1 [MI_OK] 3. MI non elaborato

Parametro: Repository	
Nome Categoria	Scelte
Membri [ME]	1. $ME < 2$ [ERROR] 2. $ME \geq 2$ [MI_OK] 3. ME non elaborato
Parametro: Repository	
Nome Categoria	Scelte
Geodispersione Community [GE]	1. GE non sufficiente [ERROR] 2. GE sufficiente [CO_OK] 3. GE non elaborato
Parametro: Repository	
Nome Categoria	Scelte
Elaborazione Conclusa [EC]	1. EC = Conclusa con errore [ERROR] 2. EC = In corso [PENDING][SINGLE] 3. EC = Conclusa con successo [EC_OK] 4. EC non elaborato
Parametro: Repository	
Nome Categoria	Scelte

Altri parametri non presi in considerazione come accesso a repository senza permesso [OT]	<ol style="list-style-type: none">1. OT sono presenti dei problemi relativi alla repository [ERROR]2. OT non ci sono problemi relativi alla repository [OT_OK]3. OT non elaborato
---	---

3.1 Funzionalità: Get Request [RF5]

Tabella 3.1: Risultati dei test per le richieste `get_request`

Test Case ID	Test Frame	Risultato
TEST_SYTE1	RA1, RH2, RE2, DF3, CO3, MI3, ME3, GE3, EC4, OT3	Errore: "Please enter a valid GitHub repository URL (https://github.com/username/repo)"
TEST_SYTE2	RA2, RH1, RE2, DF3, CO3, MI3, ME3, GE3, EC4, OT3	Errore: "Please enter a valid GitHub repository URL (https://github.com/username/repo)"

Test Case ID	Test Frame	Risultato
TEST_SYTE3	RA2, RH2, RE1, DF3, CO3, MI3, ME3, GE3, EC4, OT3	Errore: <i>"Please enter a valid GitHub repository URL (https://github.com/username/repo)"</i>
TEST_SYTE4	RA2, RH2, RE2, DF3, CO3, MI3, ME3, GE3, EC4, OT3	Successo: <i>"Request sent successfully!"</i>
TEST_SYTE5	RA2, RH2, RE2, DF1, CO3, MI3, ME3, GE3, EC4, OT3	Errore: <i>"The date must be today or a date in the past"</i>
TEST_SYTE6	RA2, RH2, RE2, DF2, CO3, MI3, ME3, GE3, EC4, OT3	Errore: <i>"The date must be today or a date in the past"</i>
TEST_SYTE7	RA2, RH2, RE2, DF3, CO2, MI2, ME2, GE2, EC3, OT2	Successo

Test Case ID	Test Frame	Risultato
TEST_SYTE8	RA2, RH2, RE2, DF3, CO1, MI2, ME2, GE2, EC1, OT2	Errore: <i>"Invalid Repository: There must be at least 100 commits!"</i>
TEST_SYTE9	RA2, RH2, RE2, DF3, CO2, MI2, ME1, GE2, EC1, OT2	Errore: <i>"Invalid Repository: Not enough members (min. 2)!"</i>
TEST_SYTE10	RA2, RH2, RE2, DF3, CO2, MI1, ME2, GE2, EC1, OT2	Errore: <i>"Invalid Repository: No milestones found (min. 1)!"</i>
TEST_SYTE11	RA2, RH2, RE2, DF3, CO2, MI2, ME2, GE1, EC1, OT2	Errore: <i>"nvalid Repository: Insufficient geographical data!"</i>
TEST_SYTE12	RA2, RH2, RE2, DF3, CO2, MI2, ME2, GE2, EC1, OT1	Errore: <i>"An Error Occurred during the analysis... Please try again later!"</i>

Test Case ID	Test Frame	Risultato
TEST_SYTE13	RA2, RH2, RE2, DF3, CO2, MI2, ME2, GE2, EC2, OT2	Successo: Visualizzazione richiesta in stato pending
TEST_SYTE14	RA2, RH2, RE2, DF3, CO2, MI2, ME2, GE2, EC3, OT2	Successo: Visualizzazione grafo rappresentante la community
TEST_SYTE15	RA2, RH2, RE2, DF3, CO2, MI2, ME2, GE2, EC3, OT2	Successo: Visualizzazione tabella community pattern
TEST_SYTE16	RA2, RH2, RE2, DF3, CO2, MI2, ME2, GE2, EC3, OT2	Successo: Visualizzazione tabella Metrics
TEST_SYTE17	RA2, RH2, RE2, DF3, CO2, MI2, ME2, GE2, EC3, OT2	Successo: Visualizzazione tabella Members

Bibliografia
