Università degli studi di Salerno

Progetto di Ingegneria del Software II

Repominer Evolution

Test Plan

Autori:
Matteo Merola
Carlo Branca
Simone Scalabrino
Giovanni Grano

Supervisore:
Prof. Andrea De Lucia

Test Plan
Versione 1.0

13 luglio 2014

Revision History

Tabella 1: Tabella delle revisioni del documento

Data	Versione	Descrizione	Autori
28/05/2014	0.9	Versione iniziale	Simone Scalabrino, Carlo Branca
29/05/2014	0.91	Aggiornamento combinazioni	Simone Scalabrino, Carlo Branca
29/05/2014	0.93	Modifica della strategia di testing	Simone Scalabrino, Carlo Branca
4/06/2014	1.0	Aggiornamento casi di test	Simone Scalabrino, Carlo Branca
4/06/2014	1.1	Modifica del testing di unità	Simone Scalabrino, Carlo Branca

Indice

1. Introduzione

Lo scopo di questo documento è quello di analizzare e gestire lo sviluppo e le attività di testing riguardanti il software RepoMinerEvo. Questa sessione di lavoro deve verificare il corretto funzionamento di RepoMinerEvo in diversi casi, studiati appositamente per mettere alla prova ogni singola funzionalità e caratteristica del sistema, al fine di ottenere un corretto funzionamento. I risultati di questi test saranno utilizzati per capire dove bisognerà intervenire, e quindi correggere eventuali errori o apportare modifiche per il miglioramento dei vari sottosistemi. Il processo verrà iterato fino a che non si otterranno i risultati attesi in accordo con i tempi di sviluppo previsti.

2. Panoramica

In riferimento al ciclo di vita del software, è chiaro osservare come, una volta che il prodotto viene rilasciato, andrà incontro per forza di cose a diverse modifiche per i più svariati motivi, dalla correzione di errori all'estensione di funzionalità. Tale concetto viene espresso dalla prima legge di Lehman dulla manutenzione del software:

un sistema deve necessariamente cambiare ed evolvere con continuità per mantenere intatto il suo livello di utilità, stante i continui cambiamenti che avvengono nell'ambiente in cui il sistema opera.

Quando un sistema cambia, per qualsivoglia motivo, la tua struttura tende a diventare più complessa. Diventa necessario utilizzare risorse supplementari per preservarne la qualità della struttura. Assume importanza estrema quindi il monitorare la qualità del codice attraverso l'utilizzo di metriche.

Nell'Ingegneria del software, la predizione dei faults è stata sempre comunemente associata a misurazioni di complessità. Il concetto di entropia più essere applicato con successo in diversi contesti, a partire dal concetto di defect prediction. L'idea di base è quella di proporre metriche complesse che sono basate sul processo di cambiamento del codice invece che sul codice stesso. Si può ragionevolemente congetturare come, un processo di cambiamento complesso e disomogeneo, possa affliggere negativamente un sistema software.

Proprio come strumento per quantificare la complessità ritorna utile il concetto di entropia di Shannon. A partire da queste informazioni è possibile andare ad implementare due modelli per la complessità dei cambiamenti sul codice; il *Basic Code Change Model* e un modello più elaborato e complesso, il *Extended Code Change Model*. Entrambi questi modelli vanno a calcolare un singolo valore che misura la complessità complessiva dei cambiamenti di un progetto durante un periodo di tempo stabilito.

Come mostrato nel caso di studio svolto nel lavoro di Hassan, tali metriche basate sulla complessità dei cambiamenti si rivelavano estremamente

valide nella defeat prediction.

Occore specificare come, in questi modelli, si andranno ad analizzare solamente le cosiddette FI, ossia *Features Introduction Modifications*; per FI intendiamo modifiche al codice che vanno ad implementare nuove features nel sistema. In questa categoria andrà a confluire tutto ciò che non può essere catalogato come FR (*Fault Repairing Modifications*) e come GM (*General Maintenance Modifications*). Come unità da misurare si è scelto il singolo file, in quanto si può ragionevolmente credere che è nel file che i developers concentrano e raggruppano le singole entità.

Come già detto, il modello BCC utilizza un arco temporale ben specificato, e definisce, file per file, la probabilità che lo stesso subisca un cambiamento. Più le probabilità sono equamente distribuite, più alta è l'entropia del sistema software, e dunque maggiore è la probabilità di svilupare difetti. Come caso limite invece, qualora un singolo file raggiunga probabilità 1, abbiamo che l'entropia del sistema diviene minima. L'idea chiave dunque risiede nel focalizzare l'analisi sulle modifiche alla singole linee di codice lungo un intervallo di tempo, allo scopo di costruire un modello di probabilità dei cambiamenti.

Quest'analisi fornisce ovviamente anche un andamento evolutivo di quella che è l'entropia del sistema lungo tutto il suo ciclo di vita. Partizionando lo stesso in periodi successivi di tempo, è interessante notare come il processo di cambiamento del codice si sia evoluto e protratto nel tempo di pari passo all'entropia dello stesso. Ciò costituisce anche un'importante parametro di monitoraggio per un manager. Si è osservato come, individuando le ragioni di picchi inaspettati nell'entropia, sia possibile pianificare ed essere pronti per problemi futuri.

Nel BCC sin qui descritto a grandi linee si assume un intervallo di tempo fissato per il calcolo dell'entropia, e si assume che il numero di file in un sistema rimanga inalterato. Ovviamente queste limitazioni restringono in campo di utilizzo di questo modello. L'*Extended Code Change* va a colmare tali lacune. In particolare, per quanto riguarda l'arco di tempo da analizzare nell'evoluzione software, è possibile prendere in considerazione diverse opzioni differenti. Abbiamo già visto il criterio del *time based periods*, ossia la suddivisione in parti eque dei periodi di tempo, a partire dall'inizio del progetto fino alla sua conclusione. Si crede che un periodo di periodo di un quarto di anno sia un limite di tempo ragionevole per raccogliere una frazione significativa nella crescita di un sistema. Un'altra possibilità riguarda i *modification limit base periods*, ossia il suddividere il tempo in periodi basandosi

sul numero di modifiche che sono memorizzare nella repository in oggetto. Per prevenire i casi in cui si verificano lunghi periodi di scarse modifiche implementative, è ragionevole imporre un limite temporale di 3 mesi qualora un periodo non raggiunga un limite fissato a 600 modifications. In conclusione, si è osservato come il processo di modifica segua delle fasi di picco, seguire da fasi di rilassamento. È sulla base di questa osservazione che è possibile la suddivisione temporale in *burst based periods*.

Nel modello ECC è necessario definire un'entropia normalizzata, con lo scopo di comparare l'entropia di sistemi di differente dimensione, con aggiunta o rimozione di files durante i vari periodi di tempo. La *normalized static entropy* H dipende dal numero di files in un sistema. È interessante notare alcuni sistemi software annoverino alcuni files con una probabilità di cambiamento davvero bassa. Per fare in modo che questi files non riducano in maniera artificiosa l'entropia del sistema, si è definita l'*adaptive sizing entropy*. L'idea di fondo è quella di dividere per il numero di linee recentemente modificate, invece per il numero attuale di linee del sistema. Vi sono due criteri di scelta per calcolare tale valore, ossia un primo criterio che prende in esame i files modificati nei precedenti x mesi,incluso il mese corrente, mentre un secondo prevede che il set di files da analizzare includa tutti i files modificati nei precedenti x periodi, incluso quello corrente.

3. Funzionalità da testare

Le componenti prese in considerazione nella fase di testing rappresentano la maggior parte delle funzionalità di RepoMinerEvo. In particolare, saranno testate:

- Metrica del numero revisioni del sistema.
- Metrica del numero medio di volte in cui i file di un package hanno subito cambiamenti.
- Metrica del numero medio di volte in cui i file di un package hanno subito operazioni di refactoring.
- Metrica del numero medio di volte in cui i file di un package hanno subito operazioni di bug fixing.
- Metrica del numero di autori di commit effettuati all'interno di un package.
- Metrica del numero di linee aggiunte o rimosse (somma, media e massimo).
- Metrica cambiamenti totali di ogni file di un dato progetto software a seguito di aggiunta di feature in un periodo di riferimento.
- Calcolo dell'entropia dei cambiamenti
- Calcolo dell'adapting size entropy

Non saranno testate le funzionalità del sistema RepoMiner originale.

4. Pass/Fail criteria

Il testing ha successo se l'output osservato è diverso dall'output atteso: ciò significa che la fase di testing avrà successo se individuerà una failure. In tal caso questa verrà analizzata e, se legata ad un fault, si procederà alla sua correzione. Sarà infine iterata la fase di testing per verificare che la modifica non abbia impattato su altri componenti del sistema.
Al contrario, il testing fallirà se l'output osservato sarà uguale all'oracolo.

5. Approccio

Nella sessione di testing di RepoMinerEvo verrà utilizzato un approccio di tipo "Category Partition", che permette di avere un sottoinsieme intelligente dei casi di test ottenibili attraverso SECT (Strong Equivalence Class Testing). Si cercherà di minimizzare le scelte delle categorie (selezionando solo quelle ritenute necessarie), in modo da evitare l'esplosione combinatorica dei casi di test.
Il testig si dividerà in due fasi distinte:

- Testing di unità, che controlla i singoli componenti (classi, metodi)
- Testing funzionale, che andrà a verificare la funzionalità dell'intero sistema assemblato

Non ci sarà una fase di testing di integrazione, dato che il sistema da realizzare è piccolo. La strategia di integrazione che si seguirà sarà di tipo big-bang.

5.1. Testing di unità

Con il testing di unità verrà effettuato un controllo delle varie classi (e dei relativi metodi) del sistema, quindi saranno ricercate le condizioni di fallimento andando ad evidenziare gli errori. Il testing di unità sarà eseguito dal team di sviluppo; potranno essere realizzate classi JUnit (per favorire il testing di regressione) o si potrà procedere ad un test manuale delle classi, qualora il tempo non sia sufficiente. Nel secondo caso, gli sviluppatori saranno liberi di scegliere alcuni dei casi di test definiti nel TCS.
Il team di testing, inoltre, provvederà alla creazione di file SQL in grado di popolare il database simulando il sistema RepoMiner. Nello specifico, sarà scritto un file SQL per ogni progetto di test individuato nel Test Case Specification.

5.2. Testing funzionale

Con il testing di sistema verrà effettuato un controllo della correttezza dell'intero sistema. Questo test sarà effettuato manualmente dai componenti del team di testing, seguendo le indicazioni presenti nel TCS, per ogni caso di test specificato.

6. Casi di test

Per sviluppare i test cases sarà utilizzato il metodo denominato Category Partition. Il metodo scelto consente di individuare semplicemente i casi di test ed evita che questi contengano combinazioni non coerenti (come WECT).

6.1. Specifica test

6.1.1. Metrica per calcolo del numero di revisioni del sistema

Parametro	Insieme dei cambiamenti	
Categoria	**Scelta**	
Numero (cnum)	1. Nessuno	[property NoChanged]
	2. Uno o più	[property Changed]

Codice	**Combinazione**	**Esito**
TC 1.1	cnum1	ok
TC 1.2	cnum2	ok

6.1.2. Metrica per il calcolo del numero medio di volte in cui i file di un package hanno subito cambiamenti

Parametro	File del package	
Categoria	**Scelta**	
Numero (fnum)	1. Zero	[property Empty]
	2. Uno o più	[property NonEmpty]
Tipo (ftype)	1. Testo	[if NonEmpty][property Text]
	2. Binario	[if NonEmpty][property Bin]
Versioning (fver)	1. Si	[if NonEmpty][property Vers]
	2. No	[if NonEmpty][property NoVers]
Parametro	Insieme dei cambiamenti	
Categoria	**Scelta**	
Numero (cnum)	1. Nessuno	[property NoChanged]
	2. Uno o più	[property Changed]

Codice	**Combinazione**	**Esito**
TC 2.1	fnum1cnum1	ok
TC 2.2	fnum1cnum2	ok
TC 2.3	fnum2ftype1fver1cnum1	ok
TC 2.4	fnum2ftype1fver2cnum1	ok
TC 2.5	fnum2ftype2fver1cnum1	ok
TC 2.6	fnum2ftype2fver2cnum1	ok
TC 2.7	fnum2ftype1fver1cnum2	ok
TC 2.8	fnum2ftype1fver2cnum2	ok
TC 2.9	fnum2ftype2fver1cnum2	ok
TC 2.10	fnum2ftype2fver2cnum2	ok

6.1.3. Metrica per il calcolo del numero medio di refactoring di un package

Parametro	File del package	
Categoria	**Scelta**	
Numero (fnum)	1. Nessuno	[property Empty]
	2. Uno o più	[property NonEmpty]
Tipo (ftype)	1. Testo	[if NonEmpty][property Text]
	2. Binario	[if NonEmpty][property Bin]
Versioning (fver)	1. Si	[if NonEmpty][property Vers]
	2. No	[if NonEmpty][property NoVers]
Parametro	Insieme dei cambiamenti	
Categoria	**Scelta**	
Numero (cnum)	1. Nessuno	[property NoChanged]
	2. Uno o più	[property Changed]
Messaggio (cmes)	1. Vuoto	[if Changed][property NoMsg]
	2. Contiene parola refactoring	[if Changed][property Refact]
	3. Non contiene parola refactoring	[if Changed][property NoRefact]

Codice	**Combinazione**	**Esito**
TC 3.1	fnum1cnum1	ok
TC 3.2	fnum1cnum2cmes1	ok
TC 3.3	fnum1cnum2cmes2	ok
TC 3.4	fnum1cnum2cmes3	ok
TC 3.5	fnum2ftype1fver1cnum1	ok
TC 3.6	fnum2ftype1fver2cnum1	ok
TC 3.7	fnum2ftype2fver1cnum1	ok
TC 3.8	fnum2ftype2fver2cnum1	ok
TC 3.9	fnum2ftype1fver1cnum2cmes1	ok
TC 3.10	fnum2ftype1fver2cnum2cmes1	ok
TC 3.11	fnum2ftype2fver1cnum2cmes1	ok
TC 3.12	fnum2ftype2fver2cnum2cmes1	ok
TC 3.13	fnum2ftype1fver1cnum2cmes2	ok
TC 3.14	fnum2ftype1fver2cnum2cmes2	ok
TC 3.15	fnum2ftype2fver1cnum2cmes2	ok
TC 3.16	fnum2ftype2fver2cnum2cmes2	ok
TC 3.17	fnum2ftype1fver2cnum2cmes3	ok
TC 3.18	fnum2ftype2fver1cnum2cmes3	ok
TC 3.19	fnum2ftype2fver2cnum2cmes3	ok
TC 3.20	fnum2ftype1fver1cnum2cmes3	ok

6.1.4. Metrica per il calcolo del numero medio di bug-fix di un package

Parametro	File del package	
Categoria	**Scelta**	
Numero (fnum)	1. Nessuno	[property Empty]
	2. Uno o più	[property NonEmpty]
Tipo (ftype)	1. Testo	[if NonEmpty][property Text]
	2. Binario	[if NonEmpty][property Bin]
Versioning (fver)	1. Sì	[if NonEmpty][property Vers]
	2. No	[if NonEmpty][property NoVers]
Parametro	Insieme di cambiamenti	
Categoria	**Scelta**	
Numero (cnum)	1. Nessuno	[property NoChanged]
	2. Uno o più	[property Changed]
Messaggio (cmes)	1. Vuoto	[if Changed]
	2. Contiene parola bug-fix	[if Changed]
	3. Non contiene parola bug-fix	[if Changed]

Codice	**Combinazione**	**Esito**
TC 4.1	fnum1cnum1	ok
TC 4.2	fnum1cnum2cmes1	ok
TC 4.3	fnum1cnum2cmes2	ok
TC 4.4	fnum1cnum2cmes3	ok
TC 4.5	fnum2ftype1fver1cnum1	ok
TC 4.6	fnum2ftype1fver2cnum1	ok
TC 4.7	fnum2ftype2fver1cnum1	ok
TC 4.8	fnum2ftype2fver2cnum1	ok
TC 4.9	fnum2ftype1fver1cnum2cmes1	ok
TC 4.10	fnum2ftype1fver1cnum2cmes2	ok
TC 4.11	fnum2ftype1fver1cnum2cmes3	ok
TC 4.12	fnum2ftype1fver2cnum2cmes1	ok
TC 4.13	fnum2ftype1fver2cnum2cmes2	ok
TC 4.14	fnum2ftype1fver2cnum2cmes3	ok
TC 4.15	fnum2ftype2fver1cnum2cmes1	ok
TC 4.16	fnum2ftype2fver1cnum2cmes2	ok
TC 4.17	fnum2ftype2fver1cnum2cmes3	ok
TC 4.18	fnum2ftype2fver2cnum2cmes1	ok
TC 4.19	fnum2ftype2fver2cnum2cmes2	ok
TC 4.20	fnum2ftype2fver2cnum2cmes3	ok

6.1.5. Metrica per il calcolo del numero di autori di commit di un package

Parametro	File del package	
Categoria	**Scelta**	
Numero (fnum)	1. Nessuno	[property Empty]
	2. Uno o più	[property NonEmpty]
Tipo (ftype)	1. Testo	[if NonEmpty][property Text]
	2. Binario	[if NonEmpty][property Bin]
Versioning (fver)	1. Sì	[if NonEmpty][property Vers]
	2. No	[if NonEmpty][property NoVers]
Parametro	Insieme di cambiamenti	
Categoria	**Scelta**	
Numero (cnum)	1. Nessuno	[property NoChanged]
	2. Uno o più	[property Changed]
Numero autori (cnaut)	1. Uno	[if Changed]
	2. Più di uno	[if Changed]
Email autori (ceaut)	1. Vuota	[if Changed]
	2. Corretta	[if Changed]

Codice	**Combinazione**	**Esito**
TC 5.1	fnum1cnum1	ok
TC 5.2	fnum1cnum2cnaut1ceaut1	ok
TC 5.3	fnum1cnum2cnaut2ceaut1	ok
TC 5.4	fnum1cnum2cnaut1ceaut2	ok
TC 5.5	fnum1cnum2cnaut2ceaut2	ok
TC 5.6	fnum2ftype1fver1cnum1	ok
TC 5.7	fnum2ftype1fver1cnum2cnaut1ceaut1	ok
TC 5.8	fnum2ftype1fver1cnum2cnaut1ceaut2	ok
TC 5.9	fnum2ftype1fver1cnum2cnaut2ceaut1	ok
TC 5.10	fnum2ftype1fver1cnum2cnaut2ceaut2	ok
TC 5.11	fnum2ftype1fver2cnum1	ok
TC 5.12	fnum2ftype1fver2cnum2cnaut1ceaut1	ok
TC 5.13	fnum2ftype1fver2cnum2cnaut1ceaut2	ok
TC 5.14	fnum2ftype1fver2cnum2cnaut2ceaut1	ok
TC 5.15	fnum2ftype1fver2cnum2cnaut2ceaut2	ok
TC 5.16	fnum2ftype2fver1cnum1	ok
TC 5.17	fnum2ftype2fver1cnum2cnaut1ceaut1	ok
TC 5.18	fnum2ftype2fver1cnum2cnaut1ceaut2	ok
TC 5.19	fnum2ftype2fver1cnum2cnaut2ceaut1	ok
TC 5.20	fnum2ftype2fver1cnum2cnaut2ceaut2	ok
TC 5.21	fnum2ftype2fver2cnum1	ok
TC 5.22	fnum2ftype2fver2cnum2cnaut1ceaut1	ok
TC 5.23	fnum2ftype2fver2cnum2cnaut1ceaut2	ok
TC 5.24	fnum2ftype2fver2cnum2cnaut2ceaut1	ok
TC 5.25	fnum2ftype2fver2cnum2cnaut2ceaut2	ok

6.1.6. Metrica per il calcolo del numero di righe aggiunte e rimosse

Parametro	File del package	
Categoria	**Scelta**	
Numero (fnum)	1. Nessuno	[property Empty]
	2. Uno o più	[property NonEmpty]
Tipo (ftype)	1. Testo	[if NonEmpty][property Text]
	2. Binario	[if NonEmpty][property Bin]
Versioning (fver)	1. Sì	[if NonEmpty][property Vers]
	2. No	[if NonEmpty][property NoVers]
Parametro	Insieme di cambiamenti	
Categoria	**Scelta**	
Numero (cnum)	1. Nessuno	[property NoChanged]
	2. Uno o più	[property Changed]
Cancellazioni (ccan)	1. Nessuna	[property NoDel]
	2. Almeno una	[if Changed and Text][property Del]
Inserimenti (cins)	1. Nessuno	[if NoChanged or Del]
	2. Almeno una	[if Changed and Vers and Text]

Codice	**Combinazione**	**Esito**
TC 6.1	fnum1cnum1ccan1cins1	ok
TC 6.2	fnum1cnum2ccan2cins1	ok
TC 6.3	fnum2ftype1fver1cnum1ccan1cins1	ok
TC 6.4	fnum2ftype1fver1cnum2ccan1cins2	ok
TC 6.5	fnum2ftype1fver1cnum2ccan2cins1	ok
TC 6.6	fnum2ftype1fver1cnum2ccan2cins2	ok
TC 6.7	fnum2ftype1fver2cnum1ccan1cins1	ok
TC 6.8	fnum2ftype1fver2cnum2ccan2cins1	ok
TC 6.9	fnum2ftype2fver1cnum1	ok
TC 6.10	fnum2ftype2fver1cnum2	ok
TC 6.11	fnum2ftype2fver2cnum1	ok
TC 6.12	fnum2ftype2fver2cnum2	ok

6.1.7. Metrica per il calcolo della dimensione media dei file modificati

Parametro	File del package	
Categoria	**Scelta**	
Numero (fnum)	1. Nessuno	[property Empty]
	2. Uno o più	[property NonEmpty]
Tipo (ftype)	1. Testo	[if NonEmpty][property Text]
	2. Binario	[if NonEmpty][property Bin]
Versioning (fver)	1. Sì	[if NonEmpty][property Vers]
	2. No	[if NonEmpty][property NoVers]
Parametro	Insieme di cambiamenti	
Categoria	**Scelta**	
Numero (cnum)	1. Nessuno	[property NoChanged]
	2. Uno o più	[property Changed]

Codice	**Combinazione**	**Esito**
TC 7.1	fnum1cnum1	ok
TC 7.2	fnum1cnum2	ok
TC 7.3	fnum2ftype1fver1cnum1	ok
TC 7.4	fnum2ftype1fver1cnum2	ok
TC 7.5	fnum2ftype1fver2cnum1	ok
TC 7.6	fnum2ftype1fver2cnum2	ok
TC 7.7	fnum2ftype2fver1cnum1	ok
TC 7.8	fnum2ftype2fver1cnum2	ok
TC 7.9	fnum2ftype2fver2cnum1	ok
TC 7.10	fnum2ftype2fver2cnum2	ok

6.1.8. Numero di cambiamenti totali di ogni file di un dato progetto (inserimento di feature)

Parametro	File del package	
Categoria	**Scelta**	
Numero (fnum)	1. Nessuno	[property Empty]
	2. Uno o più	[property NonEmpty]
Tipo (ftype)	1. Testo	[if NonEmpty][property Text]
	2. Binario	[if NonEmpty][property Bin]
Versioning (fver)	1. Si	[if NonEmpty][property Vers]
	2. No	[if NonEmpty][property NoVers]
Parametro	Insieme dei cambiamenti	
Categoria	**Scelta**	
Numero (cnum)	1. Zero	[property NoChanged]
	2. Uno o più	[property Changed]
Cancellazioni (ccan)	1. Zero	[property NoDel]
	2. Almeno uno	[if Changed and Text][property Del]
Inserimenti (cins)	1. Zero	[if NoChanged or Del][property NoIns]
	2. Almeno uno	[if Changed and Vers and Text][property Ins]
Messaggio (cmes)	1. Vuoto	[if Changed][property NoMsg]
	2. Non vuoto	[if Changed][property Msg]

Codice	**Combinazione**	**Esito**
TC 8.1	fnum1cnum1ccan1cins1	ok
TC 8.2	fnum1cnum2ccan2cins1cmes1	ok
TC 8.3	fnum1cnum2ccan2cins1cmes2	ok
TC 8.4	fnum2ftype1fver1cnum1ccan1cins1	ok
TC 8.5	fnum2ftype1fver1cnum2ccan1cins2cmes1	ok
TC 8.6	fnum2ftype1fver1cnum2ccan2cins1cmes1	ok
TC 8.7	fnum2ftype1fver1cnum2ccan2cins2cmes1	ok
TC 8.8	fnum2ftype1fver1cnum2ccan1cins2cmes2	ok
TC 8.9	fnum2ftype1fver1cnum2ccan2cins1cmes2	ok
TC 8.10	fnum2ftype1fver1cnum2ccan2cins2cmes2	ok
TC 8.11	fnum2ftype1fver2cnum1ccan1cins1	ok
TC 8.12	fnum2ftype1fver2cnum2ccan2cins1cmes1	ok
TC 8.13	fnum2ftype1fver2cnum2ccan2cins1cmes2	ok
TC 8.14	fnum2ftype2fver1cnum1	ok
TC 8.15	fnum2ftype2fver1cnum2cmes1	ok
TC 8.16	fnum2ftype2fver1cnum2cmes2	ok
TC 8.17	fnum2ftype2fver2cnum1	ok
TC 8.18	fnum2ftype2fver2cnum2cmes1	ok
TC 8.19	fnum2ftype2fver2cnum2cmes2	ok

6.1.9. Metrica per il calcolo del Basic Code Change model

Parametro	File del package	
Categoria	**Scelta**	
Numero (fnum)	1. Zero	[property Empty]
	2. Uno o più	[property NonEmpty]
Tipo (ftype)	1. Testo	[if NonEmpty][property Text]
	2. Binario	[if NonEmpty][property Bin]
Versioning (fver)	1. Si	[if NonEmpty][property Vers]
	2. No	[if NonEmpty][property NoVers]
Parametro	Insieme dei cambiamenti	
Categoria	**Scelta**	
Numero (cnum)	1. Zero	[property NoChanged][error]
	2. Uno o più	[property Changed]
Parametro	Periodo di tempo	
Categoria	**Scelta**	
Tipologia (pclass)	1. Tempo	[if Changed]
	2. Altro	[if Changed][error]
Tipo (ptype)	1. Non numerico	[if Changed][error]
	2. Numerico	[if Changed]
Lunghezza (plun)	1. ≤ 0	[if Changed][error]
	2. $>$ esistenza del sistema	[if Changed][error]
	3. Altro	

Codice	**Combinazione**	**Esito**
TC 9.1	fnum1cnum1	ok
TC 9.2	fnum1cnum2pclass1ptype1	error
TC 9.3	fnum1cnum2pclass1ptype2plun1	error
TC 9.4	fnum1cnum2pclass1ptype2plun2	error
TC 9.5	fnum1cnum2pclass1ptype2plun3	ok
TC 9.6	fnum1cnum2pclass2	error
TC 9.7	fnum2ftype1fver1cnum2pclass1ptype1	error
TC 9.8	fnum2ftype1fver1cnum2pclass1ptype2plun1	error
TC 9.9	fnum2ftype1fver1cnum2pclass1ptype2plun2	error
TC 9.10	fnum2ftype1fver1cnum2pclass1ptype2plun3	ok
TC 9.11	fnum2ftype1fver1cnum2pclass2	error
TC 9.12	fnum2ftype1fver2cnum1	error
TC 9.13	fnum2ftype1fver2cnum2pclass1ptype1	error
TC 9.14	fnum2ftype1fver2cnum2pclass1ptype2plun1	error
TC 9.15	fnum2ftype1fver2cnum2pclass1ptype2plun2	error
TC 9.16	fnum2ftype1fver2cnum2pclass1ptype2plun3	ok
TC 9.17	fnum2ftype1fver2cnum2pclass2	error
TC 9.18	fnum2ftype2fver1cnum1	error
TC 9.19	fnum2ftype2fver1cnum2pclass1ptype1	error

TC 9.20	fnum2ftype2fver1cnum2pclass1ptype2plun1	error
TC 9.21	fnum2ftype2fver1cnum2pclass1ptype2plun2	error
TC 9.22	fnum2ftype2fver1cnum2pclass1ptype2plun3	ok
TC 9.23	fnum2ftype2fver1cnum2pclass2	error
TC 9.24	fnum2ftype2fver2cnum1	error
TC 9.25	fnum2ftype2fver2cnum2pclass1ptype1	error
TC 9.26	fnum2ftype2fver2cnum2pclass1ptype2plun1	error
TC 9.27	fnum2ftype2fver2cnum2pclass1ptype2plun2	error
TC 9.28	fnum2ftype2fver2cnum2pclass1ptype2plun3	ok
TC 9.29	fnum2ftype2fver2cnum2pclass2	error

6.1.10. Metrica per il calcolo dell'Extended Code Change model

Parametro	File del package	
Categoria	**Scelta**	
Numero (fnum)	1. Nessuno	[property Empty]
	2. Uno o più	[property NonEmpty]
Tipo (ftype)	1. Testo	[if NonEmpty][property Text]
	2. Binario	[if NonEmpty][property Bin]
Versioning (fver)	1. Sì	[if NonEmpty][property Vers]
	2. No	[if NonEmpty][property NoVers]
Parametro	Insieme di cambiamenti	
Categoria	**Scelta**	
Numero (cnum)	1. Nessuno	[property NoChanged][error]
	2. Uno o più	[property Changed]
Parametro	Periodo di tempo	
Categoria	**Scelta**	
Tipologia (pclass)	1. Tempo	[if Changed][property Time]
	2. Cambiamenti	[if Changed][property Changes]
	3. Burst	[if Changed][property Burst]
	4. Altro	[if Changed][error]
Tipo (ptype)	1. Non numerico	[if Changed and not Burst][error]
	2. Numerico	[if Changed and not Burst]
Lunghezza (plun)	1. ≤ 0	[if Changed and not Burst][error]
	2. $>$ esistenza del sistema	[if Changed and Time][error]
	3. $>$ cambiamenti totali	[if Changed and Changes][error]
	4. Altro	[if Changed and not Burst]

Codice	**Combinazione**	**Esito**
TC 10.1	fnum1cnum1	ok
TC 10.2	fnum1cnum2pclass1ptype1	error
TC 10.3	fnum1cnum2pclass1ptype2plun1	error
TC 10.4	fnum1cnum2pclass1ptype2plun2	error
TC 10.5	fnum1cnum2pclass1ptype2plun4	ok
TC 10.6	fnum1cnum2pclass2ptype1	error
TC 10.7	fnum1cnum2pclass2ptype2plun1	error
TC 10.8	fnum1cnum2pclass2ptype2plun3	error
TC 10.9	fnum1cnum2pclass2ptype2plun4	ok
TC 10.10	fnum1cnum2pclass3	ok
TC 10.11	fnum1cnum2pclass4	error
TC 10.12	fnum2ftype1fver1cnum2pclass1ptype1	error
TC 10.13	fnum2ftype1fver1cnum2pclass1ptype2plun1	error
TC 10.14	fnum2ftype1fver1cnum2pclass1ptype2plun2	error
TC 10.15	fnum2ftype1fver1cnum2pclass1ptype2plun4	ok
TC 10.16	fnum2ftype1fver1cnum2pclass2ptype1	error

TC 10.17	fnum2ftype1fver1cnum2pclass2ptype2plun1	error
TC 10.18	fnum2ftype1fver1cnum2pclass2ptype2plun2	error
TC 10.19	fnum2ftype1fver1cnum2pclass2ptype2plun4	ok
TC 10.20	fnum2ftype1fver1cnum2pclass3	ok
TC 10.21	fnum2ftype1fver1cnum2pclass4	error
TC 10.22	fnum2ftype1fver2cnum1	error
TC 10.23	fnum2ftype1fver2cnum2pclass1ptype1	error
TC 10.24	fnum2ftype1fver2cnum2pclass1ptype2plun1	error
TC 10.25	fnum2ftype1fver2cnum2pclass1ptype2plun2	error
TC 10.26	fnum2ftype1fver2cnum2pclass1ptype2plun4	ok
TC 10.27	fnum2ftype1fver2cnum2pclass2ptype1	error
TC 10.28	fnum2ftype1fver2cnum2pclass2ptype2plun1	error
TC 10.29	fnum2ftype1fver2cnum2pclass2ptype2plun2	error
TC 10.30	fnum2ftype1fver2cnum2pclass2ptype2plun4	ok
TC 10.31	fnum2ftype1fver2cnum2pclass3	ok
TC 10.32	fnum2ftype1fver2cnum2pclass4	error
TC 10.33	fnum2ftype2fver1cnum1	error
TC 10.34	fnum2ftype2fver1cnum2pclass1ptype1	error
TC 10.35	fnum2ftype2fver1cnum2pclass1ptype2plun1	error
TC 10.36	fnum2ftype2fver1cnum2pclass1ptype2plun2	error
TC 10.37	fnum2ftype2fver1cnum2pclass1ptype2plun4	ok
TC 10.38	fnum2ftype2fver1cnum2pclass2ptype1	error
TC 10.39	fnum2ftype2fver1cnum2pclass2ptype2plun1	error
TC 10.40	fnum2ftype2fver1cnum2pclass2ptype2plun2	error
TC 10.41	fnum2ftype2fver1cnum2pclass2ptype2plun4	ok
TC 10.42	fnum2ftype2fver1cnum2pclass3	ok
TC 10.43	fnum2ftype2fver1cnum2pclass4	error
TC 10.44	fnum2ftype2fver2cnum1	error
TC 10.45	fnum2ftype2fver2cnum2pclass1ptype1	error
TC 10.46	fnum2ftype2fver2cnum2pclass1ptype2plun1	error
TC 10.47	fnum2ftype2fver2cnum2pclass1ptype2plun2	error
TC 10.48	fnum2ftype2fver2cnum2pclass1ptype2plun4	ok
TC 10.49	fnum2ftype2fver2cnum2pclass2ptype1	error
TC 10.50	fnum2ftype2fver2cnum2pclass2ptype2plun1	error
TC 10.51	fnum2ftype2fver2cnum2pclass2ptype2plun2	error
TC 10.52	fnum2ftype2fver2cnum2pclass2ptype2plun4	ok
TC 10.53	fnum2ftype2fver2cnum2pclass3	error
TC 10.54	fnum2ftype2fver2cnum2pclass4	error

7. Specifica dei casi di test

I test cases verranno definiti nel documento di Test Cases Specification (TCS).