

# 1 Verifica

## 1.1 Issue tracking

L'issue tracking è un'attività di supporto per i *Verificatori* e permette, oltre a tracciare potenziali errori, di assegnare i compiti di correzione al *Responsabile di progetto*.

### 1.1.1 Sintassi di una Issue

Ogni issue avrà un nome che dovrà seguire la seguente notazione:

**[D]:[T]-[S]**

dove:

- **D** rappresenta la sigla del documento di interesse;
- **T** rappresenta la tipologia di issue;
- **S** rappresenta la sintesi della descrizione del problema.

Segue, poi, la descrizione della issue. Questa dovrà obbligatoriamente contenere:

- Dove, nel documento, si trova il problema. Questo va specificato tramite sezione e, possibilmente, il numero di riga dove inizia il problema;
- Una descrizione dettagliata del problema;
- La motivazione per cui si è ritenuto necessario sollevare una issue.

## 1.2 Analisi ??

Al fine di determinare il livello di qualità del prodotto, il team deve utilizzare le tecniche di analisi statica e dinamica dei dati.

### 1.2.1 Analisi statica

Per analisi statica si intende la valutazione di un sistema basata su struttura, contenuto e documentazione senza questo venga eseguito. Questa tecnica è applicabile sia al codice, che alla documentazione stessa. L'analisi statica può avvenire con due modalità:

- *Walkthrough*: questa tecnica dev'essere applicata quando non si sa che errori o problematiche si stanno cercando. La tecnica consiste nel leggere il codice sorgente o il documento da cima in fondo per trovare anomalie di qualsiasi tipo;
- *Inspection*: questa tecnica dev'essere applicata quando si ha idea della problematica che si sta cercando e consiste in una lettura mirata del documento/codice, sulla base di una lista degli errori precedentemente stilata.

La tecnica walkthrough è molto onerosa e deve essere utilizzata soprattutto nelle prime fasi del progetto, quando non è già presente una lista degli errori comuni, oppure non si è ancora sufficientemente preparati riguardo un aspetto del progetto. Avanzando nelle fasi del progetto sarà utile stilare una lista quanto più possibile completa di errori comuni, in modo tale da evitare la tecnica walkthrough e applicare l'inspection.

### 1.2.2 Analisi dinamica

L'analisi dinamica è una forma di valutazione di un sistema software, oppure di qualche suo componente, basato sull'osservazione del suo comportamento durante l'esecuzione. Questa tecnica non è applicabile, quindi, per trovare errori nella documentazione. I test che devono essere implementati devono essere in numero relativamente ridotto e il più possibile di valore dimostrativo.

## 1.3 Verifica dei documenti

Il *Responsabile di progetto* ha il compito di dare inizio alla fase di verifica, assegnando i compiti ai *Verificatori*. Quest'ultimi, devono effettuare un'accurata verifica delle seguenti regole:

- Deve essere utilizzata una sintassi corretta e il più possibile semplice;
- Devono essere utilizzati periodi brevi;
- La struttura del documento deve essere semplice e intuitiva;
- Devono essere rispettate le norme tipografiche;
- Devono essere rispettate le regole riguardanti il glossario.

Durante l'intera attività di verifica i *Verificatori* dovranno utilizzare lo strumento del diario delle modifiche in modo tale da concentrare la loro attenzione sulle modifiche effettuate ad un documento e tenere traccia degli errori più comuni commessi nel redigere i documenti.

### 1.3.1 Sintassi

I *Verificatori* hanno il compito di identificare errori sintattici nei documenti. Ciò può essere fatto con l'ausilio di strumenti automatici, ma ogni documento deve essere sempre sottoposto a walkthrough in modo tale da individuare errori che non sono stati in grado di evidenziare. Inoltre hanno il compito di inserire nel *Glossario* tutte quelle parole che possono essere fonte di ambiguità.

### 1.3.2 Periodi

I *Verificatori* hanno il compito di calcolare l'indice Gulpease di ogni documento utilizzando lo strumento automatico preposto (vedi sezione Strumenti). Qualora non si ottenga un risultato soddisfacente il *Verificatore* deve applicare la tecnica walkthrough con l'obiettivo di individuare periodi troppo lunghi, che possono essere di difficile comprensione o leggibilità.

### 1.3.3 Struttura del documento

I *Verificatori* devono verificare che la struttura di ogni documento rispetti i contenuti del documento stesso.

## 1.4 Verifica dei diagrammi UML

I *Verificatori* devono controllare tutti i diagrammi UML prodotti, sia che venga rispettato lo standard UML, sia che siano corretti semanticamente. I diagrammi utilizzati finora sono solamente i diagrammi dei casi d'uso.

### 1.4.1 Diagrammi dei casi d'uso

Le verifiche che devono essere effettuate ai diagrammi dei casi d'uso, innanzitutto, devono riguardare il rispetto dello standard UML, soprattutto riguardanti inclusioni, generalizzazioni e estensioni. Successivamente si deve verificare che ciò che descrive il diagramma sia effettivamente il sistema che vogliamo rappresentare. Per quest'ultimo punto bisogna porre particolare attenzione sull'identificazione degli attori.

Per identificare un attore può risultare utile seguire la seguente procedura:

1. Per prima cosa è utile chiedersi se ciò che vogliamo rappresentare come attore è una persona che interagisce col sistema. In caso affermativo è giusto rappresentarlo come attore, in caso negativo si deve andare al passo 2;
2. Se ciò che vogliamo rappresentare come attore non è una persona che interagisce col sistema bisogna chiedersi se questa “cosa” può cambiare insieme al design del sistema. In caso affermativo probabilmente ciò che vogliamo rappresentare non è un attore ma una parte del sistema stesso. In caso negativo, invece, è con buona probabilità un attore.

## **1.5 Strumenti**

Da fare