

Come creare un Ticket/Bug Report (v1.0.1)

Scopo di questa pagina è descrivere come generare una segnalazione di un bug software corretta. Nel tempo la pagina si evolverà includendo eventuali nuovi esigenze e modifiche.

Introduzione

Un ticket per la segnalazione di un bug software/richiesta di assistenza deve esser **esaustivo** e **conciso**.

Chiunque, con una conoscenza di base del prodotto/software, deve esser in grado di **comprendere** la problematica segnalata e di riprodurla **nel più breve tempo possibile**.

Ciclo di vita

Il ciclo di vita di un report di un bug si può riassumere in quattro singole fasi:

1. Segnalazione
2. Validazione Segnalazione
3. Bugfixing
4. Validazione Bugfixing

In realtà il ciclo di vita di un sistema di ticketing è molto più complesso ma, per le nostre esigenze attuali, la situazione descritta di seguito è più che sufficiente.

Segnalazione

E' compito del *Segnalatore* fare in modo che un ticket risponda a **tutti** i seguenti **requisiti**:

- **Una singola segnalazione per ogni report:** Ogni singola segnalazione deve contenere il report di un solo bug; più bug devono quindi esser descritti su più report. **N.B.** Se nell'evolversi del software il comportamento della segnalazione muta drasticamente, deve nascere un nuovo report ad indicare la nuova situazione e deve esser eventualmente relazionato o chiuso quello precedente.
- **Titolo chiaro:** il titolo del report deve esser essenziale e permettere, il più possibile, di cogliere al volo la problematica segnalata.
- **Versione Software:** la versione software in cui è stata riscontrata la problematica.
- **Prodotto/Brand:** il prodotto ed il brand in cui è stata riscontrata la problematica.
- **Macchina Client:** i dati sulla macchina client utilizzata.
- **Descrizione del problema:** una descrizione sintetica ed esaustiva del problema.
- **Passi per riprodurre il problema:** una descrizione degli esatti passi da compiere, utilizzando il materiale allegato, per riprodurre il problema su una macchina differente con le stesse

componenti Hardware/Software.

- **Risultati attesi/Risultati ottenuti:** una descrizione dei risultati ottenuti e dei risultati attesi; da intendersi nel modo più generale possibile. Ad esempio, se ci si attende un particolare stile grafico bisogna sia indicare lo stile grafico che si sta vedendo in questo momento e che si considera errato, sia lo stile grafico atteso.
- **Allegati per riprodurre il problema:** il report deve esser accompagnato da una serie di allegati che permettono di:
 - **Ricreare esattamente**, in laboratorio, l'esatta problematica.
 - **Comprendere** la situazione più facilmente.
 - **Dimostrare** l'effettiva presenza di un bug.

Ad esempio il report di un bug deve esser sempre accompagnato dall'esportazione del DB per ricreare l'esatta situazione in cui è stato creato, può esser accompagnato da uno screenshot per facilitare la comprensione di una particolare schermata e può esser accompagnato dall'estratto del bus KNX per dimostrare l'assenza di alcuni tracciamenti su iKon.

- **Commentare** significa aggiungere dettagli o chiedere chiarimenti. Utilizzare i commenti per cambiare continuamente la descrizione originale della segnalazione crea confusione. **N.B.** Comportamenti differenti dalla descrizione della segnalazione implicano la creazione di nuove segnalazioni, eventualmente relazionate tra loro.

Ecco una lista consigliata di allegati utili da aggiungere alle segnalazioni:

- DB esportato del progetto (.dpadDB)
- Estratto tab Newtork di Chrome (.HAR)
- Estratto bus KNX

Chiaramente, non è possibile descrivere ogni possibile casistica; pertanto ci si affida alle capacità e alla sensibilità del Segnalatore nel descrivere nel migliore dei modi il report di un bug, seguendo queste linee guida.

Per qualsiasi approfondimento è sufficiente documentarsi un attimo: esistono migliaia di articoli e libri in merito. Ecco qualche esempio:

- <http://www.chiark.greenend.org.uk/~sgtatham/bugs.html>
- <http://noverse.com/blog/2012/06/how-to-write-a-good-bug-report/>

Estratto Network di Chrome (.HAR)

E' buona regola effettuare il test con lo strumento di analisi di rete aperto, offerto da Chrome. Per usarlo è sufficiente premere, CTRL+Shift+I avviando gli strumenti di sviluppo, e poi selezionando il tab "Nerwork" In caso di problemi, premendo il tasto destro del mouse sulla lista, è possibile salvare un estratto .HAR.

Il file estratto può esser gestito utilizzando diversi tool come:

- <http://www.softwareishard.com/har/viewer/>
- <http://ericduran.github.io/chromeHAR/>

Validazione Segnalazione

E' compito del *Validatore* rivedere tutti i report segnalati dal *Segnalatore*.

In fase di validazione è necessario:

- **Controllare il report:** la segnalazione deve rispondere ai requisiti in modo esaustivo; in caso contrario è necessario richiedere al Segnalatore di correggere, aggiungere o confermare eventuali cambiamenti.
- **Validare la presenza del bug nella versione software desiderata:** è necessario ripercorrere la segnalazione e controllare che, nella versione del software da correggere, il baco sia realmente presente nelle modalità descritte dal *Segnalatore*. L'operazione di validazione della presenza del bug è necessaria perchè permette non solo di confermare la presenza del problema ma anche di comprendere se il report contiene realmente tutti i dati necessari per esser replicato in fase di bugfixing.
- **Girare il ticket:** a validazione completata, se l'operazione ha avuto esito positivo in tutti i suoi aspetti, il *Validatore* gira il ticket allo *Sviluppatore*; in caso contrario lo indirizza al *Segnalatore* per chiedere ulteriori dettagli o conferme.

Bugfixing

TODO

Validazione Bugfixing

TODO

From:

<http://www.domotalabs.com/dokuwiki/> - **DOMOTICA LABS WIKI**

Permanent link:

<http://www.domotalabs.com/dokuwiki/doku.php?id=faqs:howtocreteaticket&rev=1444300058> 

Last update: **2015/10/08 11:27**