

Clean Coder
Robert C. Martin

Jacopo De Angelis

29 settembre 2020

Indice

1	Professionalità	5
1.1	Prendere le proprie responsabilità	5
1.2	Non fare del male	5
1.2.1	Non danneggiare la funzionalità	5
1.2.2	I QA non dovrebbero trovare niente	6
1.2.3	Devi sapere che funziona	6
1.2.4	QA automatizzato	7

Capitolo 1

Professionalità

1.1 Prendere le proprie responsabilità

Come si può "imparare" a prendersi le proprie responsabilità? Ci sono alcuni principi da seguire.

1.2 Non fare del male

Come può un programmatore fare del male? Può farlo al software, creando problemi per le funzioni e alla struttura del software.

1.2.1 Non danneggiare la funzionalità

Chiaramente il software deve funzionare, non solo per noi programmatori ma anche per clienti e datori di lavoro. Per essere professionali, insomma, non bisogna creare bug. Il concetto è che quello deve essere l'impegno di un programmatore e nel caso escano dei bug deve prendersene la responsabilità se causati dal proprio codice.



Figura 1.1: Tu in questo momento

1.2.2 I QA non dovrebbero trovare niente

Qual è il codice per il quale non sai se i QA¹ troveranno qualcosa?
Il codice di cui non si è certi.

Usare i QA come dei cacciatori di bug rende il processo più lungo e riduce la fiducia nei programmatori. Mandare in testing codice di cui non si è sicuri è violare la regola del "non fare del male".

I QA troveranno errori nel codice? Probabile.

1.2.3 Devi sapere che funziona

Come? Semplice, testando il codice.

Paura di metterci troppo? Automatizzali scrivendo degli unit test.

Quanto codice andrebbe testato? Tutto.

¹Quality assurance, coloro incaricati di testare il codice

1.2. NON FARE DEL MALE

7

1.2.4 QA automatizzato