

How Do You Refactor Untested Code?

Quotes

Posted by:

Posted on : 2010/2/9 9:51:38

I am currently reading the excellent "Debug It!" book written by Paul Butcher and I wanted to share with you some of the little gems that I have found in it.

"Bug fixing often uncovers opportunities for refactoring. The very fact that you're working with code that contains a bug indicates that there is a chance that it could be clearer or better structured."

"As a rule of thumb, for every user who tells you about a problem, there will be between 10 and 100 other users who experienced the same problem and didn't think to get in touch."

"Once you start to get on top of your quality issues, you're going to want to start refactoring the old, cruffy, untested code. And you should - the point of exercise is to clean up problems, and refactoring is a key element of that process. Remember, however, that refactoring crucially depends upon the support of an extensive suite of automated tests. Without tests, you're not refactoring. You're hacking. So, how do you refactor untested code? You don't. The first thing you do is to write the tests."

Source: "Debug It!", Paul Butcher, Pragmatic Bookshelf, 214 pages