

Użyj pokrycia, aby ulepszyć swoje testy jednostkowe

Emily Bache



Mierzenie pokrycia testami jest łatwiejsze niż kiedykolwiek. W nowoczesnym IDE przycisk do uruchamiania testów pokrycia znajduje się tuż obok tych, które umożliwiają ich wykonanie lub debugowanie. Wyniki pokrycia są prezentowane klasa po klasie wraz z małymi wykresami, zaś w kodzie źródłowym odpowiednie wiersze są podświetlane kolorem.

Dane dotyczące pokrycia są łatwe do zdobycia. Jak najlepiej z nich skorzystać?

Gdy piszesz nowy kod

Większość osób zgadza się, że razem z całym napisanym kodem należy dostarczać testy jednostkowe. Możesz spierać się o to, w jakiej kolejności robić te rzeczy, ale z mojego doświadczenia wynika, że najlepiej sprawdzają się krótkie pętle sprzężenia zwrotnego. Utwórz mały kod testowy, napisz trochę kodu produkcyjnego i zbuduj funkcjonalność wraz z testami. Gdy pracuję w ten sposób, od czasu do czasu przeprowadzam testy pokrycia jako dodatkowe ubezpieczenie, że nie zapomniałam przetestować nowego kodu, który właśnie napisałam.

Głównym zagrożeniem jest tutaj to, że możesz być bardzo zadowolony z wysokiej wartości pokrycia i nie zauważysz, że brakuje ci zarówno kodu, jak i testów dla kluczowej funkcjonalności. Być może zapomniałeś dodać obsługę błędów. Być może przegapiłeś regułę biznesową. Jeśli nigdy wcześniej nie napisałeś kodu produkcyjnego, pomiary pokrycia nie mogą tego wykryć.

Gdy musisz zmienić kod, którego nie napisałeś

Modyfikowanie kodu, którego nie napisałeś i dla którego testy są słabe lub ich brakuje, może być trudne – szczególnie jeśli tak naprawdę nie rozumiesz, co on robi, ale mimo to musisz go zmienić. Gdy mam do czynienia z taką sytuacją, sprawdzenie pokrycia jest jednym ze sposobów, w jaki dowiaduję się, jak dobre są testy i które części z większą pewnością mogą refaktoryzować.