

Fragen, Antworten und Kommentare zur aktuellen Vorlesung

Antwort: Für MC/DC muss in jedem Booleschen Ausdruck jeder Term einmal diesen Ausdruck „geschaltet“ haben. Dies bedeutet, dass es für jeden Term zwei Tests gibt, bei denen der betrachtete Term einmal false und einmal true ist, weiterhin müssen die anderen Terme immer den gleichen Booleschen Wert haben und der Gesamtausdruck einmal nach false und einmal nach true ausgewertet werden. Solch eine Auswertung muss es, gerade bei Kurzschlussauswertungen nicht immer geben.

Im Beispiel gehören 1,2 für **a**, 3,4 für **b** und 4,5 für **c** zusammen, es werden also höchstens 6 Testfälle benötigt, es können aber wie im Beispiel ausreichen. Im Beispiel haben die Variablen immer den gleichen Wert wie das Ergebnis, das muss nicht sein, die zugehörigen Ergebnisse (pro Farbe) müssen sich nur unterscheiden.

Nr.	a	b	c	a & (b c)
1	true	true	true	true
2	false	true	true	false
3	true	true	false	true
4	true	false	false	false
5	true	false	true	true

Es gibt in der Literatur Stellen, in denen das Umschalten auch für Boolesche Teilterme gefordert wird. Dies wäre in den Beispieldaten mit den Testfällen 3 und 4 auch der Fall, da a konstant true ist und (b|c) einmal false und einmal true ist und dann das Ergebnis auch einmal true und einmal false ist.

Frage: Wie kann ich sicherstellen, dass ich während der gesamten Entwicklung ausreichend teste?

Antwort: Es geht zunächst nicht nur um das Testen sondern um begleitende QS-Maßnahmen während jedes Entwicklungsschritts. Die passenden Maßnahmen hängen von vielen Faktoren ab, z. B. der Komplexität und Wichtigkeit der Aufgabe, aber auch vom potenziellen Schaden bei Fehlern. Die Arbeit in Paaren oder gemeinsame Durchsprachen von Text-Dokumenten in Teams sind sehr hilfreich. Bei der Erstellung von Anforderungen, unabhängig vom gewählten Ansatz, ist immer über die Test- und Prüfbarkeit nachzudenken. Dabei können grobe Testbeschreibungen parallel zu den Anforderungen entstehen.

In eigentlichen Entwicklungsphasen sollten immer Tests geschrieben werden, die auch anderen Personen zur Verfügung stehen. Frühzeitig sind Tests zu erstellen, von denen angenommen wird, dass sie langfristig im Projekt genutzt werden.

Frage: Wie kann TDD in einem Projekt eingeführt werden?

Antwort: Generell stellt sich die Frage, ob alle entwickelnden Personen den Ansatz nutzen müssen oder das Vorgehen individuell bleiben soll. Dies ist keine fachliche Frage, da aber nicht alle Personen Änderungen wollen und vielleicht auch benötigen, ist der Weg der Einführung vorher zu klären. Bevor TDD eingeführt wird, ist es wichtig, dass alle die Grundlagen und Vorteile von TDD verstehen. Der Kern von TDD besteht darin, zuerst Tests zu schreiben und dann den Code, der diese Tests bestehen lässt. Beginnen Sie damit, Tests für kleine, überschaubare Teile zu schreiben. Führen Sie die Tests oft aus, idealerweise jedes Mal, wenn Sie eine Änderung am Code vornehmen. Dies hilft, Fehler frühzeitig zu erkennen und zu beheben. Nachdem Code existiert, der den Test besteht, sollten Sie

den Code überarbeiten (Refactoring) und verbessern, ohne das Verhalten zu ändern. Führen Sie nach jedem Refactoring die Tests erneut aus, um sicherzustellen, dass alles noch wie erwartet funktioniert. TDD ist eine Fähigkeit, die mit der Praxis verbessert wird. Lernen Sie aus persönlichen Erfahrungen und Erfahrungen anderer und suchen Sie ständig nach Möglichkeiten, den TDD-Prozess zu verbessern.