SQM, 2.5.22

Frage: Typischerweise werden Testfälle doch nach dem Arrange-Act-Assert-Ansatz geschrieben, ist das on Go nicht der Fall?

Antwort: Der Ansatz ist in Go genauso nutzbar, erst das zu testende Ausgangsszenario basteln (Arrange), dann die zu testende Funktionalität ausführen (Act) und dann mit Zusicherungen prüfen ob genau das gewünschte Ergebnis erreicht wurde (Assert). Es gilt als üblich, dass ein Testfall dann nicht weitergeht, da es bei entdeckten Fehlern keinen Sinn macht. Der Go-Ansatz geht etwas weiter und erlaubt es, dass Tests weitergehen, auch wenn ein Assert scheitert, was für einen Testfall abgefragt werden kann. Die Motivation besteht darin, dass die Person, die die Tests entwickelt entscheiden kann, ob es nach dem gefundenen Fehler noch sinnvoll ist den Test weiterlaufen zu lassen oder nicht. Dies macht pragmatisch bei stark zustandsbasierten Systemen einen Sinn, da sonst für jeden Schritt ein neuer Test geschrieben werden muss, was sich kritisch auf die Test-Ausführungszeit auswirken kann. Dies interessante Idee ist in JUnit leider nur als schmutziger Work-Around durchführbar.

Frage: Anscheinend unterstützt Go nur die Anweisungsüberdeckung?

Antwort: Für die offizielle Go-Testumgebung ist das leider der Fall. Diese bewusst getroffene Entscheidung ist auf den Go-Seiten dokumentiert. Aus meiner Sicht zeigt die Vorlesung, dass die Entscheidung bewusst falsch ist, da selbst Überdeckungen nur ein hinreichendes Testmittel sind.