

## Fragen, Antworten, Kommentare und Hinweise

Diskussion: Erfahrungen mit Prozessen in Unternehmen

In kleineren Unternehmen findet oft keine explizite Prozessmodellierung statt, da alle Beteiligten die Abläufe kennen oder sich durch direkte Rückfragen informieren. Bei vielen Prozessen in der Software-Entwicklung findet eine implizite Prozessmodellierung statt, wenn z. B. Scrum als konsequentes Vorgehensmodell genutzt wird. Eine weitere Prozessgestaltung findet durch die Auswahl und Konfiguration von Werkzeugen statt, wenn z. B. das Vorgehen mit Arbeitsaufträgen und Entwicklungsschritten in Confluence, Jira, Jenkins, Wiki oder einem anderen Werkzeug festgelegt wird.


Ein explizite Prozessmodellierung macht die Prozesse sichtbar und kann so Diskussionen und Ideen für Optimierungen anstoßen.

Frage: Wer macht das QM für die QM-Prozesse?

Antwort: Ein Ansatz ist es, dass es die QM-Abteilung selbst macht und sich z. B. bezüglich gegebener Vorgaben oder gesetzter Ziele orientiert. Dazu werden auch Vertreter der durch das QM betreuten Prozesse befragt. Der zweite, auch ergänzend nutzbare, Ansatz ist es, die QM-Analyse extern z. B. von einem Beratungsunternehmen durchführen zu lassen. Auf diesen Weg kann externes Wissen in den Prozess gelangen. Ist das Unternehmen zertifiziert, z. B. ISO 9001, findet eine regelmäßige Überprüfung statt, bei der das QM des QM eine wesentliche Rolle spielt.

Frage: Warum erhalte ich durch einen einfachen import in Go einen Fehler?

Antwort:



```

2 package main
3
4 import (
5     "fmt"
6     "math/rand"
7     "time"
8 )

```

Build Ausgabe




C:/kleukersSEU/go/bin/go.exe build [F:/workspace:  
# Sleepsort  
.\main.go:5:2: imported and not used: "fmt"  
**Fehler: Prozess beendet mit Rückgabewert 2.**

Dies passiert, wenn nichts aus dem Paket im aktuellen Programm genutzt wird. Dahinter ist eine durchaus sinnvolle Default-Compiler-Einstellung, die aber ggfls. nerven kann. Generell können importe umbenannt werden, ein Trick dabei ist der Name `_` wodurch die oben genannte Überprüfung entfällt. Sollte dann `fmt` genutzt werden sollen, ist der Unterstrich zu löschen. Pakete sind hier auch umbenennbar, statt `_` ein String ohne Hochkommata, das ist aber nur in Extremfällen sinnvoll.

Nebenbei wird `_` eigentlich genutzt, wenn ein Paket nur wegen seiner Seiteneffekte, also z. B. Variablen, aber nicht Funktionen genutzt wird, siehe z. B.

<https://stackoverflow.com/questions/21220077/what-does-an-underscore-in-front-of-an-import-statement-mean> .

```
2 package main
3
4 import (
5     "fmt"
6     "math/rand"
7     "time"
8 )
```

Build Ausgabe   

**C:/kleukersSEU/go/bin/go.exe build [F:/works**  
**Erfolg: Prozess beendet mit Rückgabewert 0.**