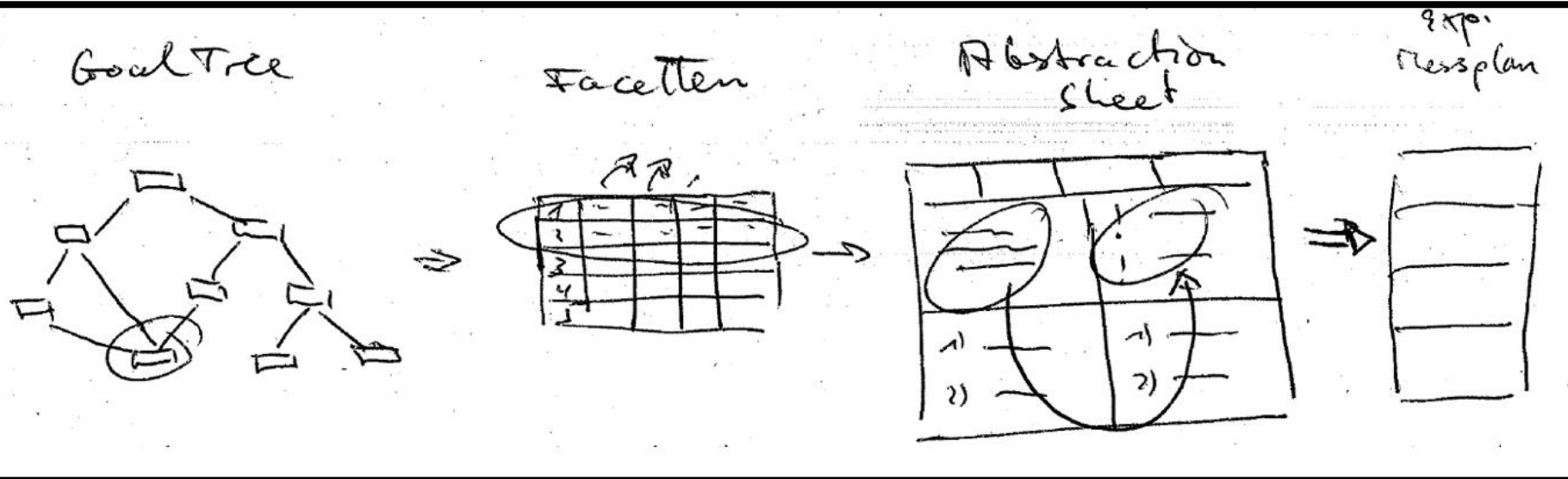


# Übung: Software-Qualität

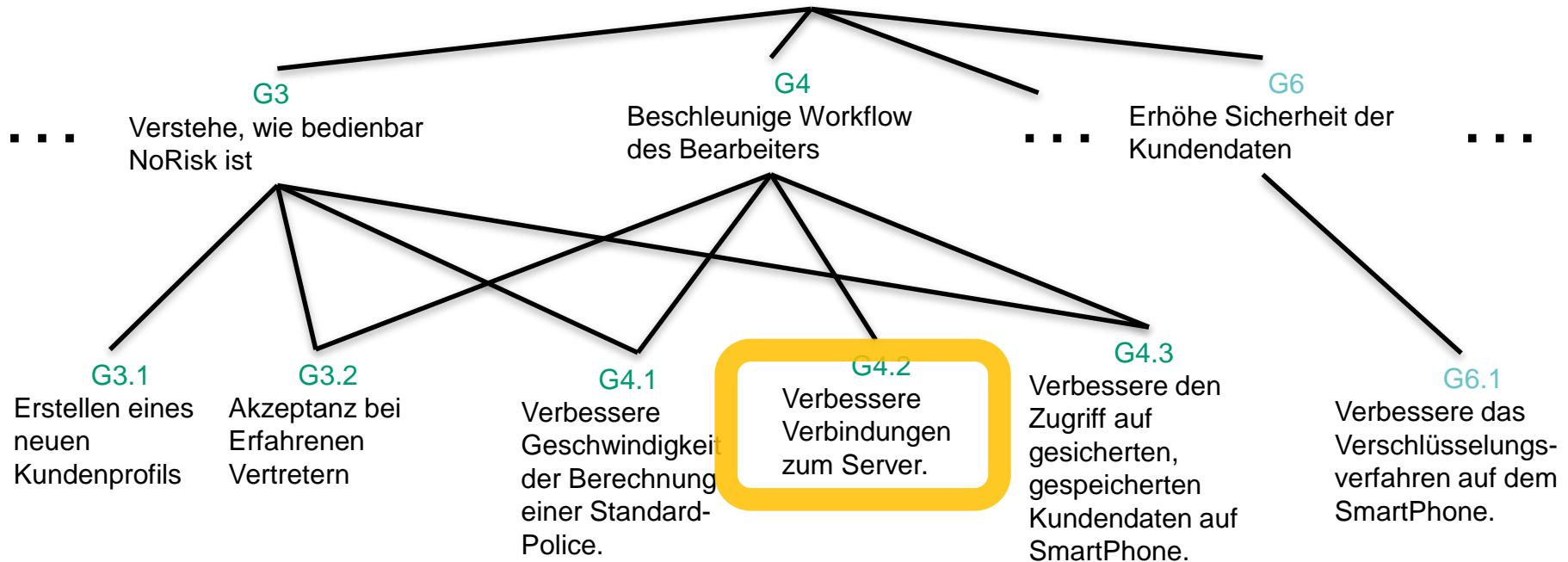
Sommersemester 2016

[swq@se.uni-hannover.de](mailto:swq@se.uni-hannover.de)



- 1) Ziel (mit Kunden) festlegen, konkretisieren: Goal-Tree
- 2) Ziel aus verschiedenen Perspektiven betrachten: Facetten
- 3) Herausfinden, **was** das Ziel beeinflusst: AbstractionSheet
- 4) Question, Metrik und Messplan extrahieren: Experimentelle Überprüfung

## Verbessere NoRisk



**Beachte:** GQM kann sowohl auf ein Produkt angewendet werden (wie im Beispiel) als auch auf Prozesse, wie z.B. im Softwareentwicklungsprojekt.

## 2. Schritt: Hauptziel verfeinern mit Zielfacetten

- Ziele im Zielbaum noch abstrakt, ungenau
- Zielfacetten fragen „gezielt“ nach den wichtigsten Informationen eines Ziels:

Was soll getan werden?

- Untersuche
- Verbessere
- ...

Welcher Q-Aspekt wird betrachtet?

Was soll untersucht werden?

Aus welcher Sicht soll gemessen werden?

Ziel	Zweck	Qualitätsaspekt	Betrachtungsgegenst.	Perspektive
1.1	Verbessere	Speichernutzung	Quellcode	Entwickler
1.2	Verbessere	Komplexität	Quellcode	Entwickler
...	...	...	...	...
4.1	Verbessere	Geschwindigkeit	Berechn StandardPolice	Bearbeiter
4.2				
...	...	...	...	...

## 2. Schritt: Hauptziel verfeinern mit Zielfacetten

- Ziele im Zielbaum noch abstrakt, ungenau
- Zielfacetten fragen „gezielt“ nach den wichtigsten Informationen eines Ziels:

Was soll getan werden?

- Untersuche
- Verbessere
- ...

Welcher Q-Aspekt wird betrachtet?

Was soll untersucht werden?

Aus welcher Sicht soll gemessen werden?

Ziel	Zweck	Qualitätsaspekt	Betrachtungsgegenst.	Perspektive
1.1	Verbessere	Speichernutzung	Quellcode	Entwickler
1.2	Verbessere	Komplexität	Quellcode	Entwickler
...	...	...	...	...
4.1	Verbessere	Geschwindigkeit	Berechn. StandardPolice	Bearbeiter
4.2	Verbessere	Geschwindigkeit	Verbind. zum Server	Bearbeiter
...	...	...	...	...

# Abstraction Sheet

Zweck Verbessere	Qualitätsaspekt Geschwindigkeit	Betrachtungsgegenstand Verbindung zum Server	Perspektive u. Umgebung Bearbeiter
---------------------	------------------------------------	---	---------------------------------------

## Qualitätsaspekt konkretisiert: Q-Faktoren

Welche Faktoren definieren den Qualitätsaspekt?

- Dauer des Aufbaus der gesicherten Verbindung und der Übertragung einer Standard-Police
- subjektives Aufhalte/Unterbrechungs-Empfinden des Bearbeiters (Workflow)

## Einflussfaktoren

Was hat Einfluss auf die Qualitätsfaktoren?

- verwendete Mobilfunktechnologie
  - Hintergrundläufigkeit des Übertragungsprozesses
- Wenn an diesen E-Faktoren „gedreht“ wird, dann beeinflusst das den Q-Aspekt! → Stichwort: Dreh-Rädchen!

## Ausgangshypothese:

Momentane Erwartung über die Qualitätsfaktoren?

- Der Aufbau einer gesicherten Funk-Verbindung zum Server dauert über das Mobilfunknetz (GSM) 15 Sek.
- Die Übertragung einer verschlüsselten Standard-Police dauert danach 30 Sek. .
- Der Bearbeiter muss während dieser Zeit auf NoRisk warten und wird in seinem Workflow unterbrochen, der Kunde wird nervös.

## Einflusshypothese:

Was beeinflusst wie die Qualitätsfaktoren?

- Die verwendete Mobilfunktechnologie beeinflusst die Übertragungsgeschwindigkeit, z.B.: Übertragung bei GSM sollte schneller sein als bei EDGE
- Das subjektive Warte-Empfinden wird beeinflusst, wenn Aktionen im Hintergrund erledigt werden und dabei nicht die Hauptanwendung aufhalten.

# McCabe-Wert

Abgeleitet aus dem „Programmablaufgraph G“

$$V(G) = e - n + 2$$

e = edges, n = nodes

## Algorithmus

gilt für strukt. Sprachen (Pascal, Cobol)

$$\begin{aligned} V(G) &= \text{Anzahl (IFs, Schleifen)} \\ &+ \text{CASE-Fälle (je Anw. -1)} \\ &+ 1 \end{aligned}$$

## Fehler-Risiko abschätzen

$V(G) > 10$  : mittel

$V(G) > 20$  : hoch

$V(G) > 50$  : unbeherrschbar

# McCabe-Wert berechnen

- Berechnen Sie anhand des Codes den McCabe-Wert von folgender Methode:

```
public static int getTriangleKind(int sideA, int sideB, int sideC){  
    if(sideA == sideB){  
        return (sideB == sideC) ? 3 : 2;  
    } else if(sideA == sideC || sideB == sideC){  
        return 2;  
    } else {  
        return 1;  
    }  
}
```



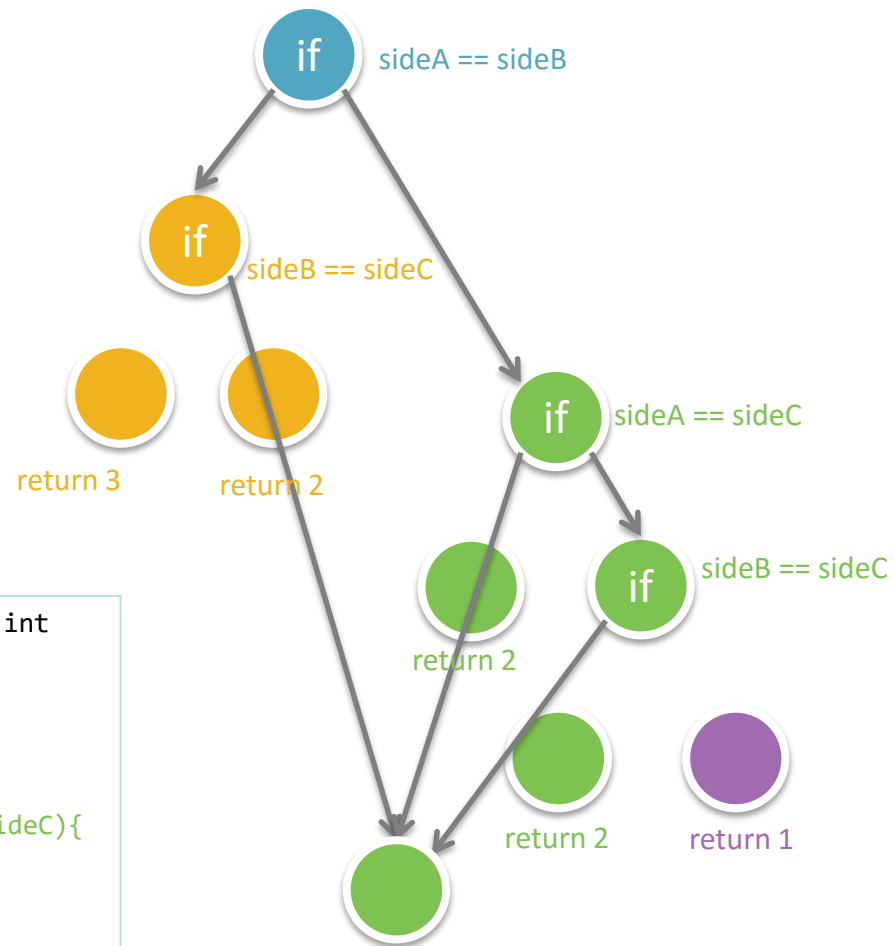
# McCabe-Wert berechnen

- Berechnen Sie nun den McCabe-Wert unter Verwendung des Programmablaufgraphen.

```
public static int getTriangleKind(int sideA, int sideB, int sideC){  
    if(sideA == sideB){  
        return (sideB == sideC) ? 3 : 2;  
    } else if(sideA == sideC || sideB == sideC){  
        return 2;  
    } else {  
        return 1;  
    }  
}
```

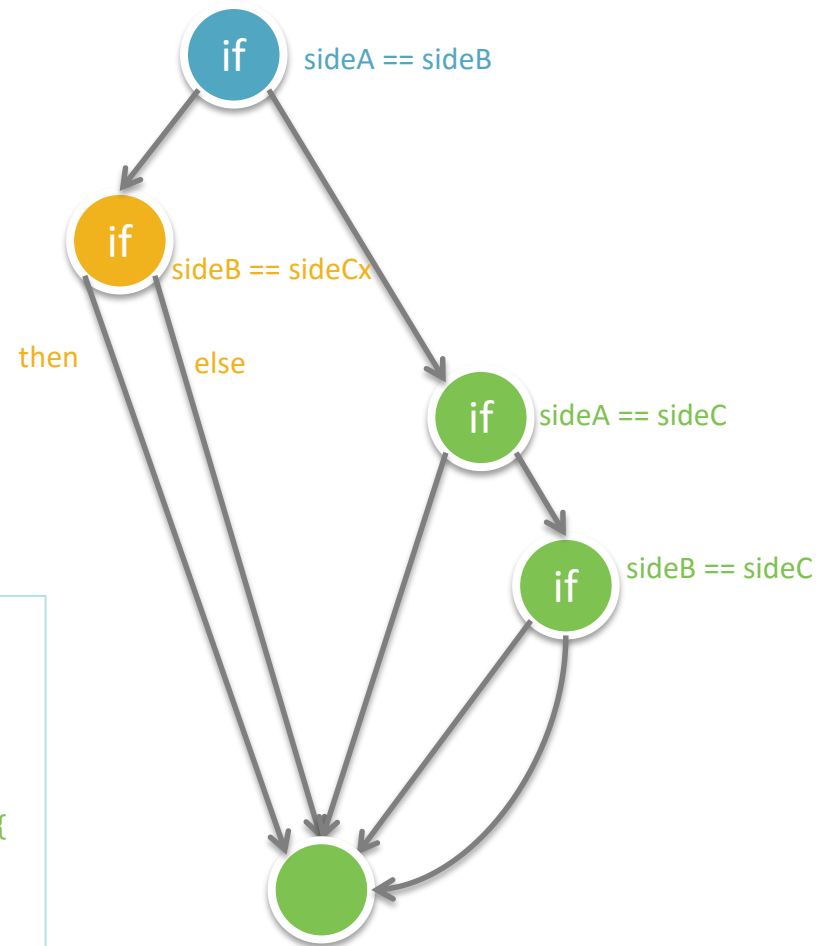
# McCabe-Wert berechnen

```
public static int getTriangleKind(int sideA, int sideB, int sideC){  
    if(sideA == sideB){  
        return (sideB == sideC) ? 3 : 2;  
    } else if(sideA == sideC || sideB == sideC){  
        return 2;  
    } else {  
        return 1;  
    }  
}
```



# McCabe-Wert berechnen

```
public static int getTriangleKind(int sideA, int
sideB, int sideC){
    if(sideA == sideB){
        return (sideB == sideC) ? 3 : 2;
    } else if(sideA == sideC || sideB == sideC){
        return 2;
    } else {
        return 1;
    }
}
```



$$8E - 5N + 2$$