

# Konsent statt Konsens

Wie Teams schneller zu tragfähigen  
Entscheidungen kommen!



## Patrick Loka (M.Sc. Informatik)

- Senior Software Engineer
- Schwerpunkte
  - Software-Architektur
  - Migration und Integration
  - Web Engineering
  - Mobile
  - Theoretische Informatik
- Projektarbeit, Consulting, Trainings
- Vorträge, Workshops, Artikel



## Daniel Krämer (M.Sc. Informatik)

- Senior Software Engineer
- Schwerpunkte
  - Software-Architektur
  - Migration und Integration
  - Web Engineering
  - Clean Code
  - Testautomatisierung
- Trainings, Vorträge, Artikel





## Individuelle Anwendungsentwicklung - Java Enterprise, Web, Mobile

- seit 2005 ♦ in Köln ♦ für alle Branchen ♦ **Goldschmiede**  anderScore
- nach Aufwand & zum Festpreis
- ✓ Digitalisierung / Prozesse / Integration
- ✓ Migration
- ✓ Neuentwicklung
- ✓ Notfall / kritische Situation
- ➔ pragmatisch, zielgerichtet, zuverlässig

## Kompletter SW Life Cycle

- Projektmanagement / agile Methodik
- Anforderungsanalyse
- Architektur & SW-Design
- Implementierung & Testautomation
- Studien & Seminare



**... und für Sie? Sprechen Sie uns an!**

**umsetzbar**

**bedarfsorientiert**

**von allen getragen**

**funktionsfähig**

**ASAP**

**dokumentiert**

**nachvollziehbar**

## Das Team entscheidet!



Fullstack



Senior



DevOps



Architekt



Security

**Was?**

**Wie?**

# Was entscheidet das Team?



**Stakeholder**



**Fullstack**



**Senior**



**DevOps**



**Architekt**



**Security**



**Product Owner**



**Betrieb**



**Anderes Team**

## Monolith vs. Microservices



Fullstack

**Microservices:**

*“Das ist cool!”*



Senior

**Monolith:**

*“Das hat doch schon immer funktioniert.”*



DevOps

**Monolith:**

*“Kein Know How, keine Infrastruktur keine Microservices.”*



Architekt

**Microservices:**

*“Das passt am Besten zu den Anforderungen.”*



Security

**Egal:**

*“Hauptsache sicher!”*



## Anarchie: Survival of the fittest!



Fullstack

**Microservices:**

*“Das ist cool!”*



Senior

**Monolith:**

*“Das hat doch schon immer funktioniert.”*



DevOps

**Monolith:**

*“Kein Know How, keine Infrastruktur keine Microservices.”*



Architekt

**Microservices:**

*“Das passt am Besten zu den Anforderungen.”*



Security

**Egal:**

*“Hauptsache sicher!”*

## Autokratie: Der Teamleader entscheidet!



Fullstack

**Microservices:**

*“Das ist cool!”*



Senior

**Monolith:**

*“Das hat doch schon immer funktioniert.”*



DevOps

**Monolith:**

*“Kein Know How, keine Infrastruktur keine Microservices.”*



Architekt

**Microservices:**

*“Das passt am Besten zu den Anforderungen.”*



Security

**Egal:**

*“Hauptsache sicher!”*

## Demokratie: Der Mehrheitsbeschaffer entscheidet!



Fullstack

**Microservices:**

*“Das ist cool!”*



Senior

**Monolith:**

*“Das hat doch schon immer funktioniert.”*



DevOps

**Monolith:**

*“Kein Know How, keine Infrastruktur keine Microservices.”*



Architekt

**Microservices:**

*“Das passt am Besten zu den Anforderungen.”*



Security

**Egal:**

*“Hauptsache sicher!”*

## Im Konsens: Es wird nichts entschieden!



Fullstack

**Microservices:**

*“Das ist cool!”*



Senior

**Monolith:**

*“Das hat doch schon immer funktioniert.”*



DevOps

**Monolith:**

*“Kein Know How, keine Infrastruktur keine Microservices.”*



Architekt

**Microservices:**

*“Das passt am Besten zu den Anforderungen.”*



Security

**Egal:**

*“Hauptsache sicher!”*

## Lieblingslösung



*“So würde ich  
entscheiden.”*

## Inakzeptabel



*“Ich kenne eine  
bessere Lösung!”*

**Alle sind dafür!**



## Lieblingslösung



*“So würde ich  
entscheiden.”*

## Akzeptabel



*“Das wird auch  
funktionieren.”*

## Inakzeptabel



*“Das wird nicht  
funktionieren,  
**weil...**”*

# Niemand ist dagegen!

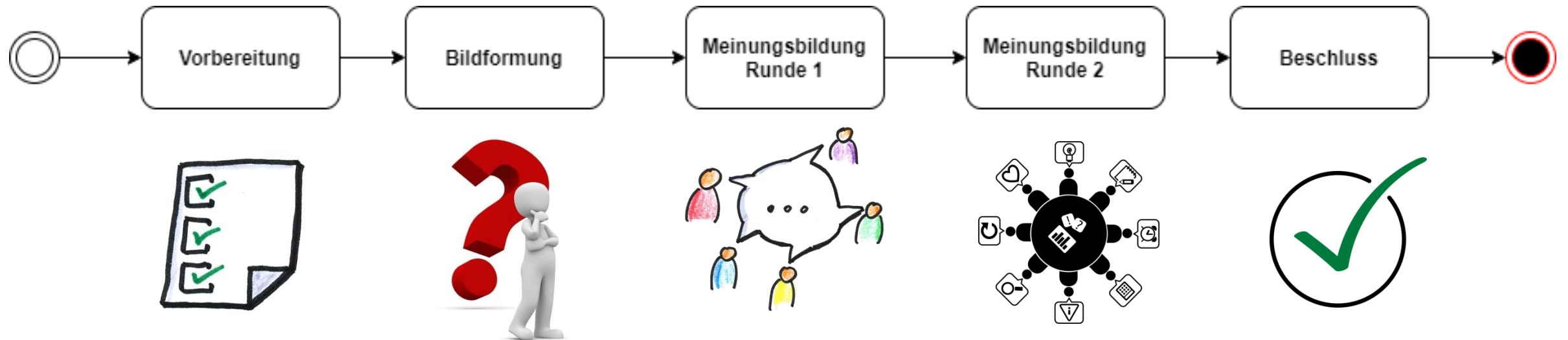


## Hintergrund:

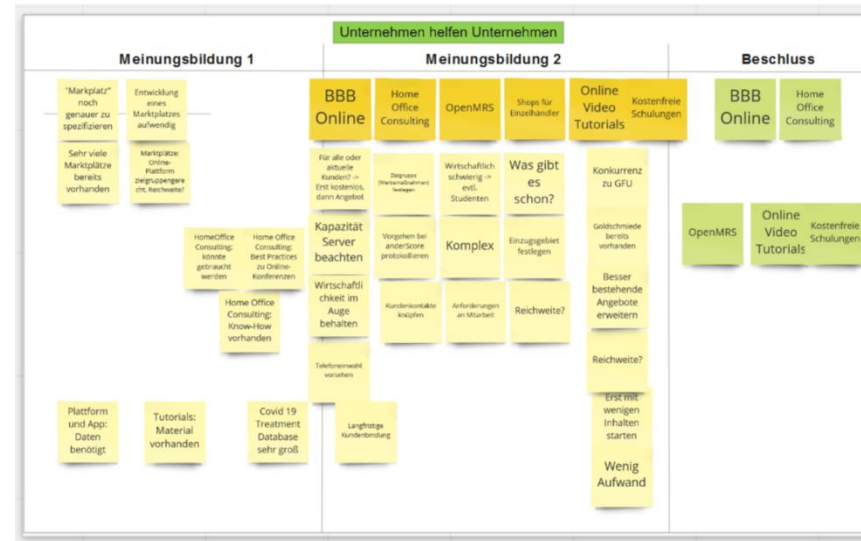
- Socius (lat.): *Gefährte*, kratein (griech.): *regieren*
- Methode zur Entscheidungsfindung (und Organisationsstruktur...)
- Wegbereiter: Comte (1851), Boeke (1945), Endenburg (1971)

## Philosophie:

- Bedenken im Fokus (Konsent statt Konsens)
- Argumente statt Positionen
- Stärkung der gemeinsamen Ausrichtung auf das Ziel



- Teilnehmer über Thema und Vorgehen informieren
- Optional: freiwillige Teilnahme
- Moderator benennen
  - **Möglichst** kein inhaltlicher Beitrag
  - Agiles Team: Scrum Master
- Material vorbereiten
  - Whiteboard
  - Slicky Notes





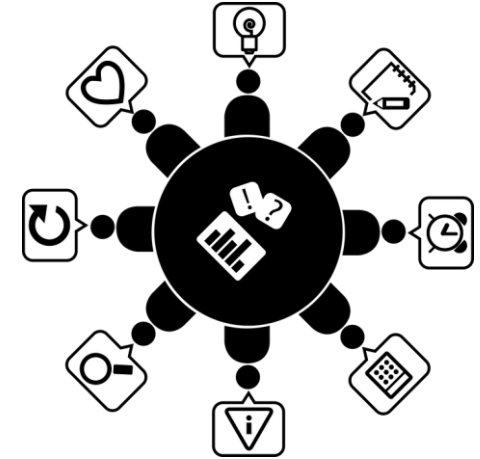
- Vorstellung des Themas (Themeninhaber)
  - Was ist der Hintergrund des Meetings?
  - Worum geht es thematisch konkret?
  - Worin soll das Ziel bestehen?
- Klärung von Fragen (Teilnehmer)
  - Habe ich alles richtig verstanden?
  - Welche Informationen benötige ich zur Meinungsbildung?



- Welche Meinung vertrete ich?
- Was habe ich mir vorab überlegt?
- Welche Ideen kommen mir gerade in den Sinn?
- Moderator
  - Teilnehmer der Reihe nach dran nehmen
  - Ideen auf Sticky Notes festhalten



- Wie stehe ich zu dem Gehörten?
  - Möchte ich noch etwas ergänzen?
  - Kommen mir nun noch weitere Ideen?
  - Welche Meinung vertrete ich **jetzt**?
- 
- Moderator
    - Clustering der Sticky Notes
    - Benennung der Cluster



- Moderator: Präsentation eines Lösungsvorschlages
- Konsentabfrage
  - Konsent erreicht: Verkünden der Entscheidung
  - Konsent nicht erreicht: Iterative Annäherung in weiterer Runde
- Dokumentation
  - Fotografieren des Whiteboards
  - Kommunikation des Beschlusses an Stakeholder



- Hierarchie vor Fachkompetenz
  - Teilnehmer sind gleichberechtigt
- Überstimmung von Minderheiten
  - Schwerwiegende Bedenken bewirken ein Veto
- Der Lauteste gewinnt
  - Moderator entscheidet, wo die Runde beginnt
  - Jeder kommt der Reihe nach zu Wort
- Langwierige Diskussionen
  - Moderator nimmt Personen einzeln dran



## Im Konsent: Bedenkenorientierte Entscheidung!

Konsentabfrage:  
“Microservices”



Fullstack

**Microservices:**

*“Das ist cool!”*



Senior

**Monolith:**

*“Das hat doch schon immer funktioniert.”*



DevOps

**Monolith:**

*“Kein Know How, keine Infrastruktur keine Microservices.”*



Architekt

**Microservices:**

*“Das passt am Besten zu den Anforderungen.”*



Security

**Egal:**

*“Hauptsache sicher!”*

- Thema (Richtungsentscheidung!) klar abgrenzen
- Nicht mehr als 7 Teilnehmer vorsehen
- Externen Moderator benennen
- Inhaltlich und didaktisch vorbereiten

- Phasen konsequent einhalten
- Fragen zu Beginn stellen
- Aspekte auf den Punkt bringen
- Sachlich bleiben
- Diskussionen unterbinden
- Offtopic bei Bedarf gesondert erfassen
- Time Box nicht überschreiten

- Bei Zeitverzug: Folgemeeting planen
- Ergebnisse nicht nachverhandeln
- Bei nachträglichen Bedenken Folgemeeting ansetzen

# Vielen Dank für ihr Interesse!

## Gerne beantworten wir noch ihre Fragen...



<https://github.com/anderscore-gmbh/semicolon-21.12>

© 2021 anderScore GmbH

15.10.2020

Patrick Loka, Daniel Krämer