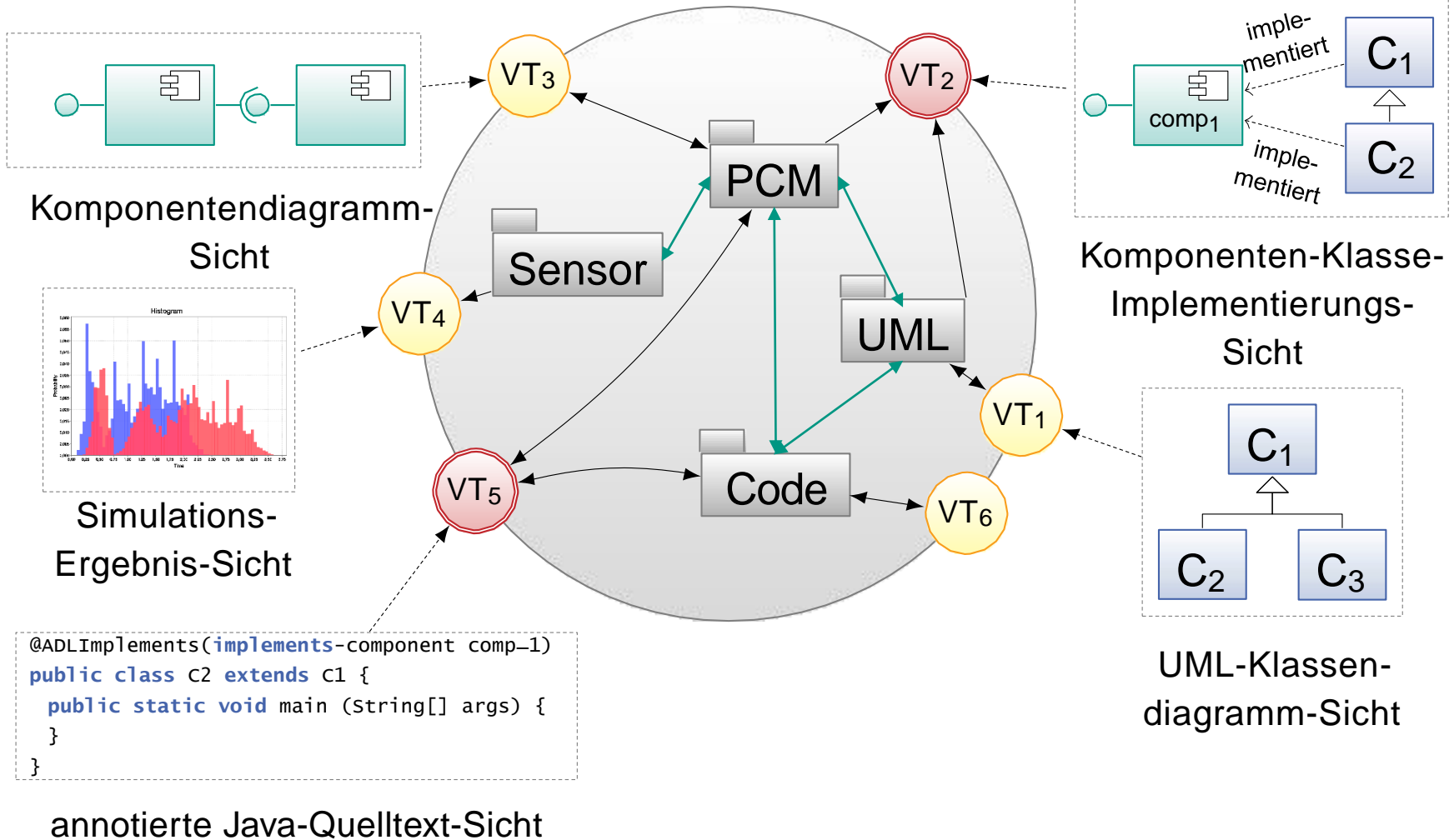


Betreuer: Georg Hinkel, Erik Burger

# Inkrementelle Modellsichten



# Sichtenbasierte Entwicklung



# Beispiel-Sichtdefinition mit ModelJoin

```
import ...
target "http://sdq.ipd.kit.edu/mdsd/ComponentSpeed/0.2"

theta join TimeSpanSensor with AssemblyContext
where "TimeSpanSensor.sensorName.indexOf(AssemblyContext.id) > 0"
as AssemblyContext {
    keep attributes NamedElement.entityName, Sensor.sensorName
    keep outgoing encapsulatedComponent__AssemblyContext as type
Component {
    ...
}
keep incoming Experiment.sensors as type Experiment {
    ...
}
}
```

Erneute Ausführung nach jeder Modelländerung dauert lange.  
Beschleunigung durch inkrementelle Verarbeitung wünschenswert.

# Inkrementalisierungsframeworks

- Am KIT/FZI entwickelte Frameworks können Analysen und Transformationen implizit inkrementell ausführen
  - Queries
  - Transformationen
  - Erweiterbar für beliebige Ausdrücke
- **Offene Fragen:**
  - Lassen sich Sichtdefinitionen generisch auf Transformationen abbilden?
  - Welche Verbesserungen hinsichtlich Skalierbarkeit lassen sich dadurch erzielen?
  - Lassen sich in NMF eingebaute Programminvertierungen anwenden?

# Inkrementelle Modellsichten

## Problem

- Sichtinstanzen müssen bei jeder Modelländerung neu erzeugt werden
- Schlechte Skalierbarkeit bei großen Modellen

## Idee

- Inkrementelle Propagation von Updates
- Nutzung von vorhandenen Frameworks für Inkrementalisierung

## Ansatz

- Mapping von ModelJoin auf Inkrementelle Ausführungsumgebung von NMF
- Validierung durch Fallstudien

