Test-Driven Scenario Specification of Automotive Software Components

Carsten Wiecher, Joel Greenyer, Jan Korte

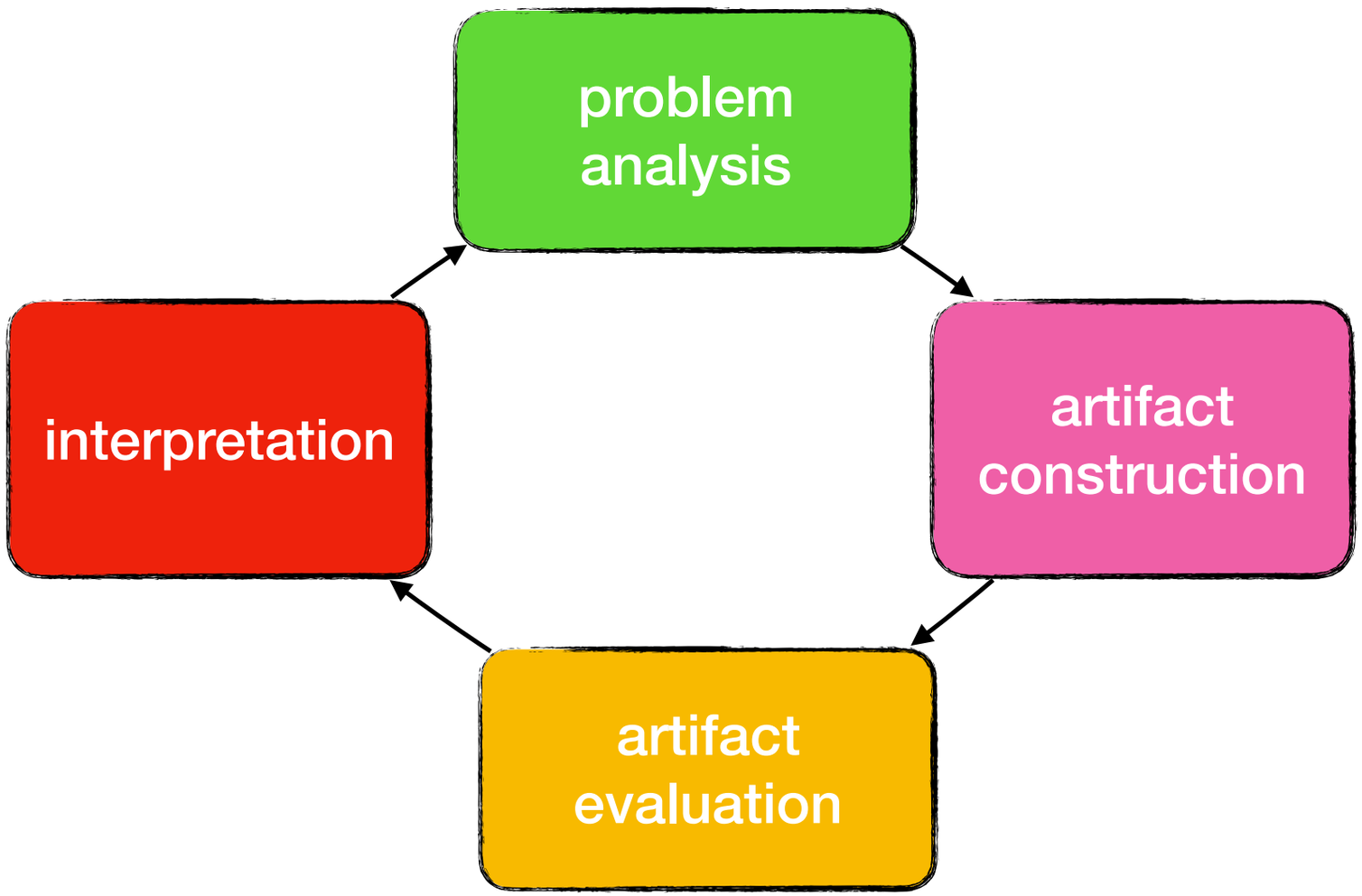
Carsten Wiecher

- **6 years job experience at Leopold Kostal GmbH & Co. KG as software engineer**
- **since 12/2018 research associate at IDiAL Institute - FH Dortmund**
- **phd student at Paderborn University & FH Dortmund**

carsten.wiecher@fh-dortmund.de

Research Context and Design

Design Science Research
-building and evaluating a solution-
-assessing its actual impact in reality-



UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

KOSTAL

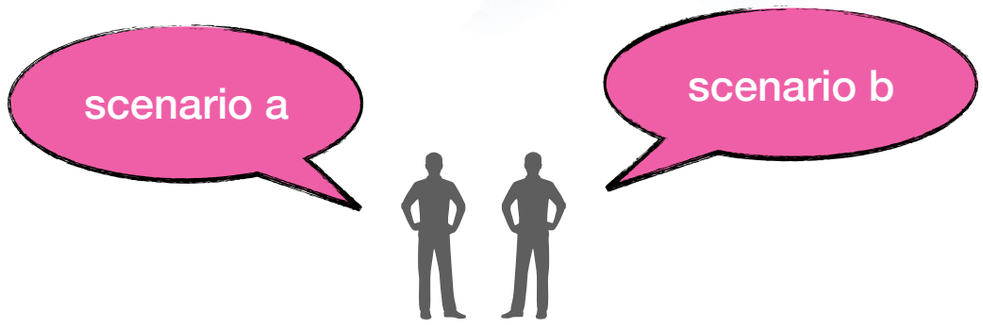
Fachhochschule Dortmund
University of Applied Sciences and Arts

Problem Description and Motivation for TDSS

complex automotive software systems
A. high contextual complexity
B. high dynamics in projects

→ costly development iterations
→ high verification/validation effort

How to support a natural and intuitive formalization of requirements with the goal to:
A. automatically analyze these requirements?
B. systematically work out test cases based on these requirements?



→ reduce development costs by:
A. early detection of requirement inconsistencies and ambiguities
B. reusing test cases in later process steps

Example: Onboard Charger for electric vehicles

Req1: No temperature related derating of the available output power shall be commanded, if the coolant inlet temperature sensor reads values between -40°C and 65°C.

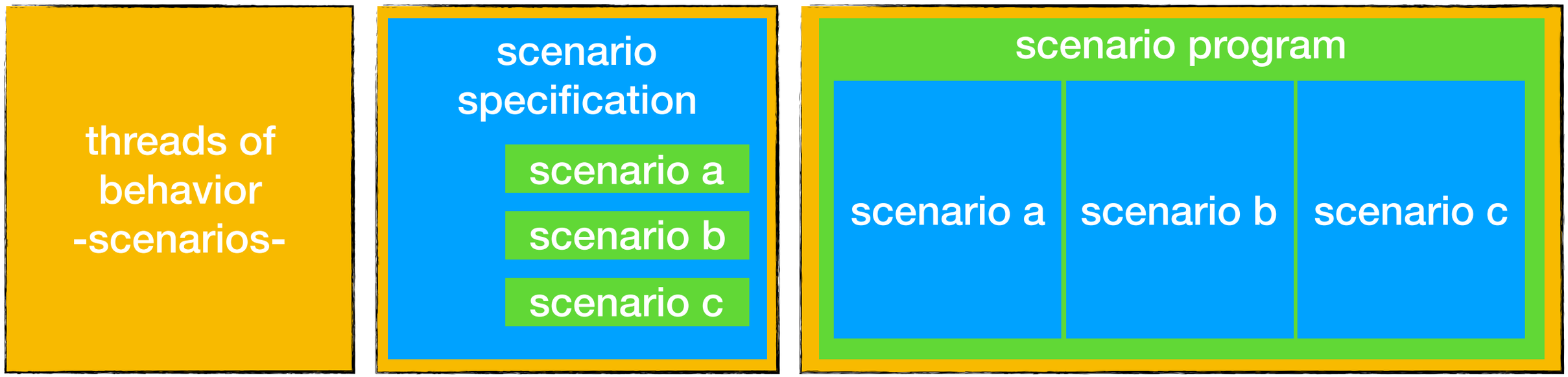
Req2: If the coolant inlet temperature sensor reads values between 65°C and 75°C, linear derating with 1/10 of maximal output power per 1°C shall be commanded.

Req3: Power-down above 75°C coolant inlet temperature.

Req4: Power-down if coolant inlet temperature increases more than 5°C within 5s.

Req5: Power-down if PCB temperature increases more than 20°C within 3s.

Background: Scenario Modeling Language for Kotlin (SMLK)

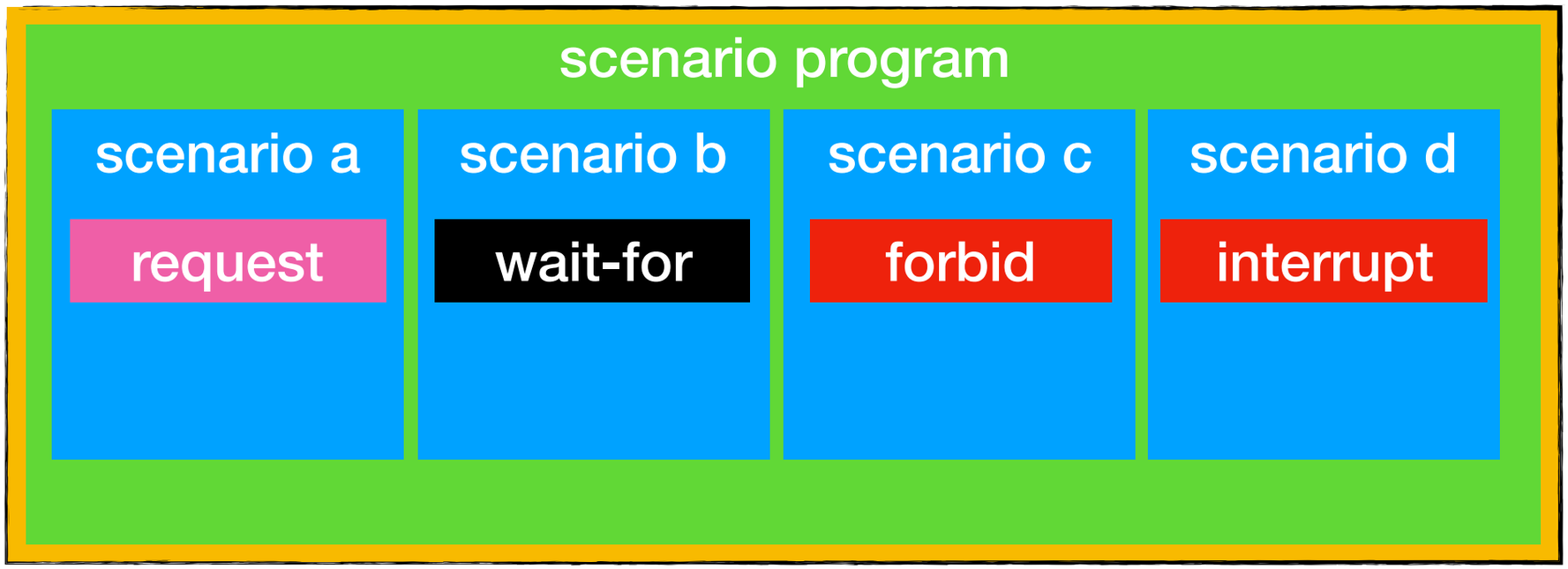


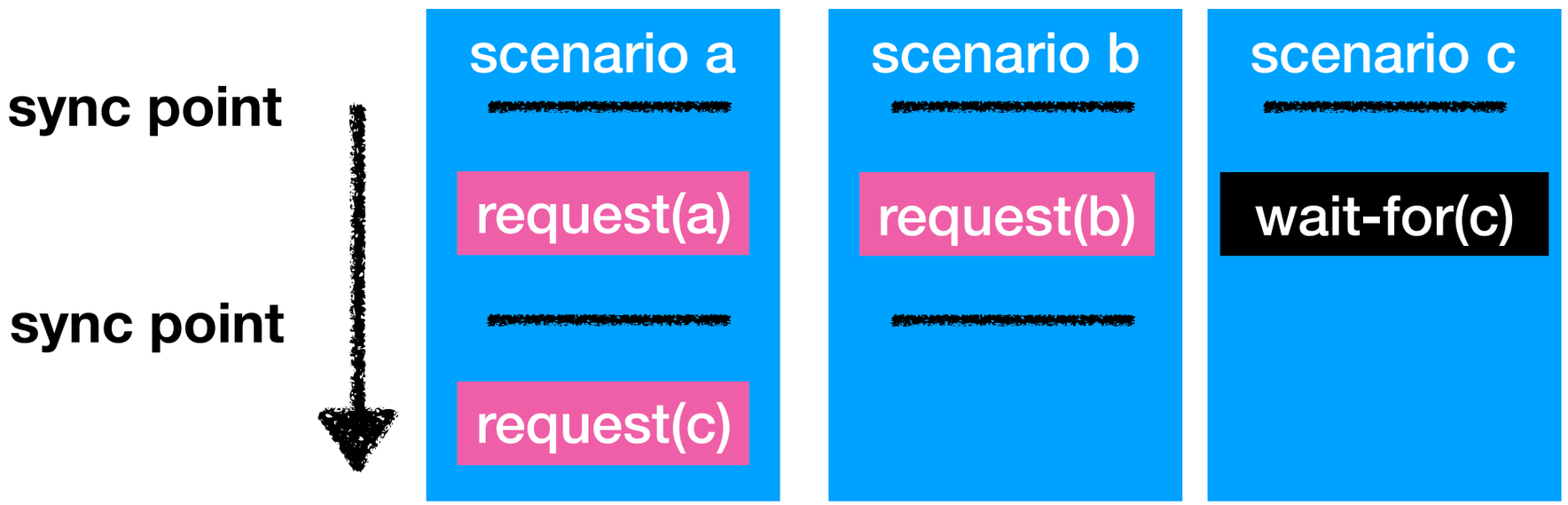
Req1: No temperature related derating of the available output power shall be commanded, if the coolant inlet temperature sensor reads values between -40°C and 65°C.

```
scenario(deratingComponent.startCycle){
    if (deratingComponent.coolantTemp in -40..65)
        request(deratingComponent.setDeratingFactor(1.0))}
```

Behavioral Programming: http://www.wisdom.weizmann.ac.il/~bprogram/
Live Sequence Charts: http://wiki.weizmann.ac.il/playgo/index.php/Live_sequence_charts
Scenario Modeling Language: http://scenariotools.org/scenario-modeling-language/

Background: Scenario Modeling Language for Kotlin (SMLK)

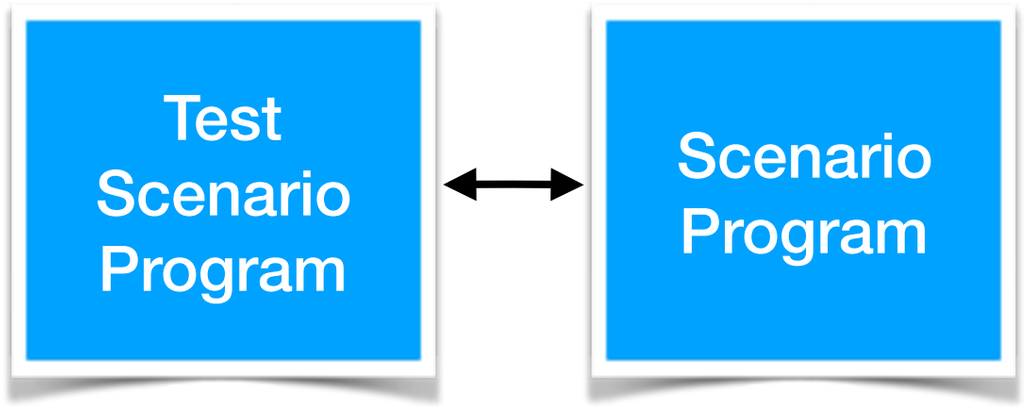




Test-Driven Scenario Specification of Automotive Software Components (TDSS)

request external events (e.g. temperature values)

wait for scenario program events (e.g. expected derating factor)



request, wait-for, forbid, interrupt specific events to model the behavior of the software component

Test-Driven Scenario Specification of Automotive Software Components (TDSS)

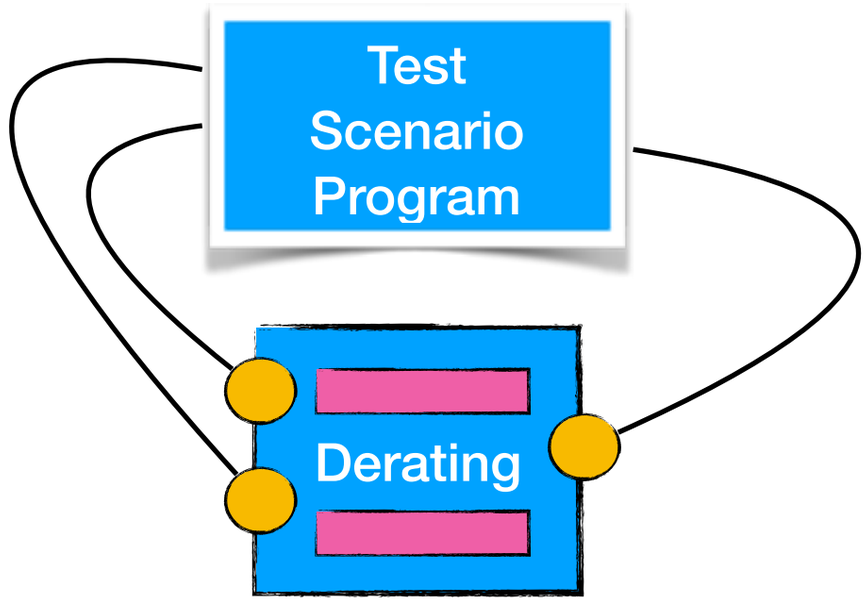
```
@Test
fun `At 66°C the DeratingFactor is 0-9`() =
    runTest(deratingScenarioProgram){
     deratingIOSequence(Triple(66, 66, 0.9))}
```

request(temperature values)

request(start cycle)

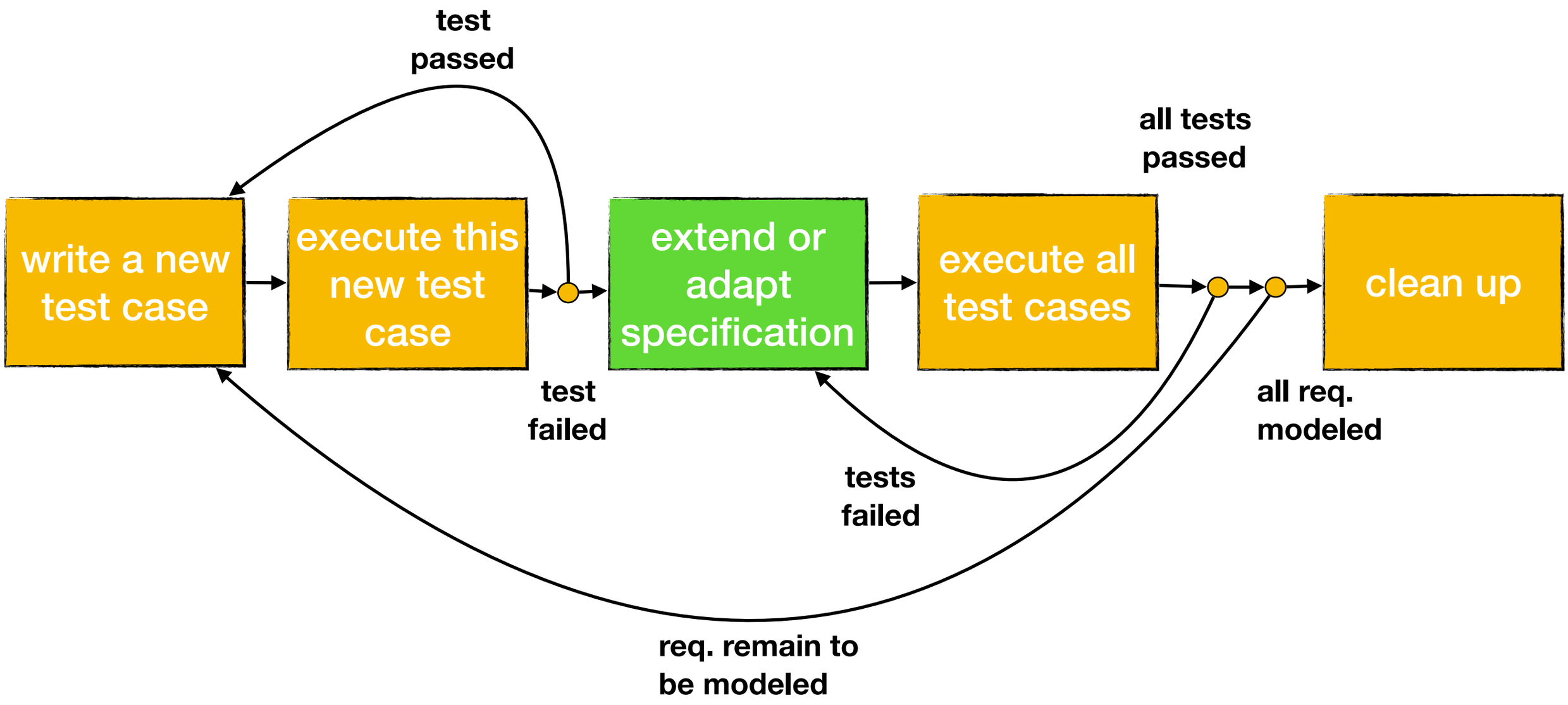
waitFor(derating value)

waitFor(end cycle)

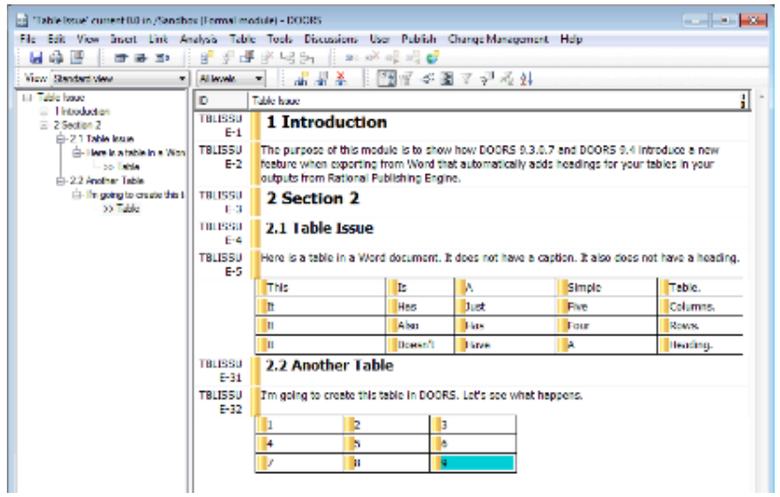


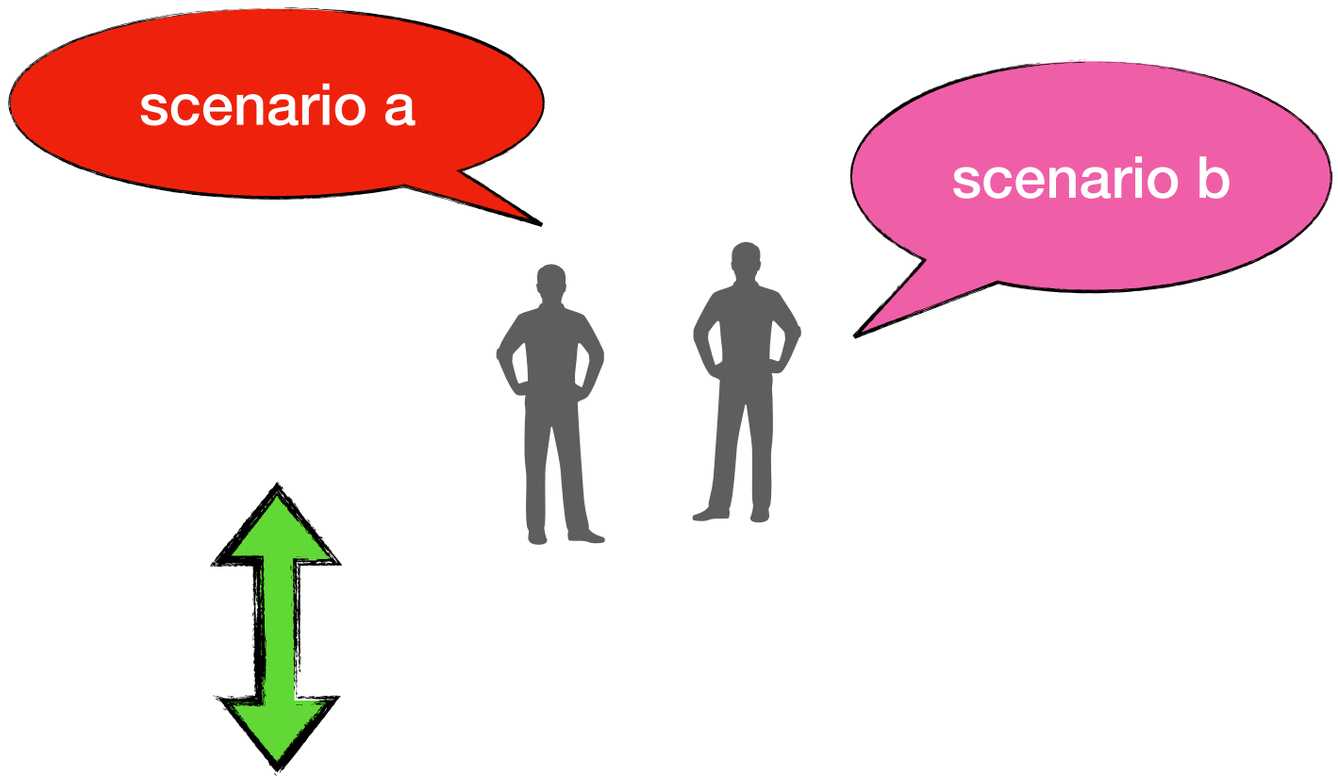

```
cycleScenario(deratingComponent){
    if (deratingComponent.coolantTemp in 65..75)
        request(deratingComponent.setDeratingFactor(
   (75.0-deratingComponent.coolantTemp)/10))}
```

Test-Driven Scenario Specification of Automotive Software Components (TDSS)



Test-Driven Scenario Specification of Automotive Software Components (TDSS)






```
cycleScenario(deratingComponent){
    if (deratingComponent.coolantTemp in 65..75)
        request(deratingComponent.setDeratingFactor(
    (75.0-deratingComponent.coolantTemp)/10))
}
```

TDSS - Evaluation

IBM DOORS

Software Requirements Analysis

Req1: No temperature related derating of the available output power shall be commanded, if the coolant inlet temperature sensor reads values between -40°C and 65°C.

Req4: Power-down if coolant inlet temperature increases more than 5°C within 5s.

Software Qualification Test

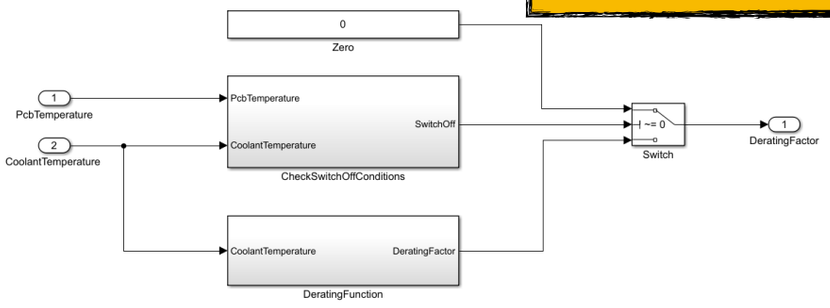
Software Architectural Design

Software Integration and Integration Test

MATLAB

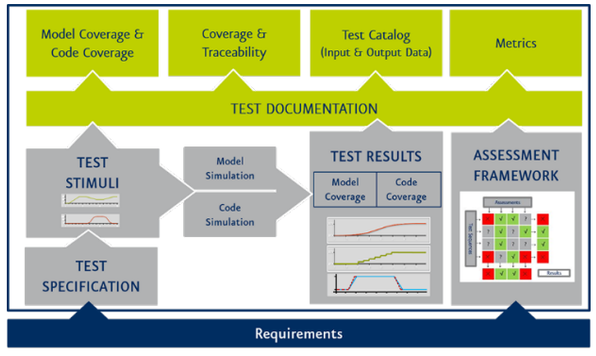
Software Detailed Design and Unit Construction

Software Unit Verification



Assessment Result	Test001/TSeq001	Test001/TSeq002	Test001/TSeq003	Test001/TSeq004
mars_RS1 : Failed				
mars_RS2 : Failed				
mars_RS3 : Passed				
mars_RS4 : Passed				
mars_RS5 : Passed				

Assessment Result	Test001/TSeq001	Test001/TSeq002	Test001/TSeq003	Test001/TSeq004
mars_RS1 : Passed				
mars_RS2 : Passed				
mars_RS3 : Passed				
mars_RS4 : Passed				
mars_RS5 : Passed				



TDSS - Summary

Test-Driven Scenario Specification of Automotive Software Components:

- **new approach for the test-driven, scenario-based requirements specification**
- **combines agile practices with formal specification and analysis**
- **SMLK is suited to formalize functional requirements in an intuitive way**
- **immediately testing of requirements:**
  - **feeling of control to the requirements specification and analysis phase**
  - **high confidence of correctness when the final implementation is tested: tests vs. requirements vs. implementation**

Outlook

1. Combinatorial Test Design
2. Transition to Design
3. Transition to Validation
4. Evaluation: based on DSR framework, systematically assessing the impact in reality