2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

MUNICH, GERMANY
OCTOBER 14-15, 2025

Automated Flow to Maintaining Consistency in Parallel Design Representations Using Cross-Level Verification

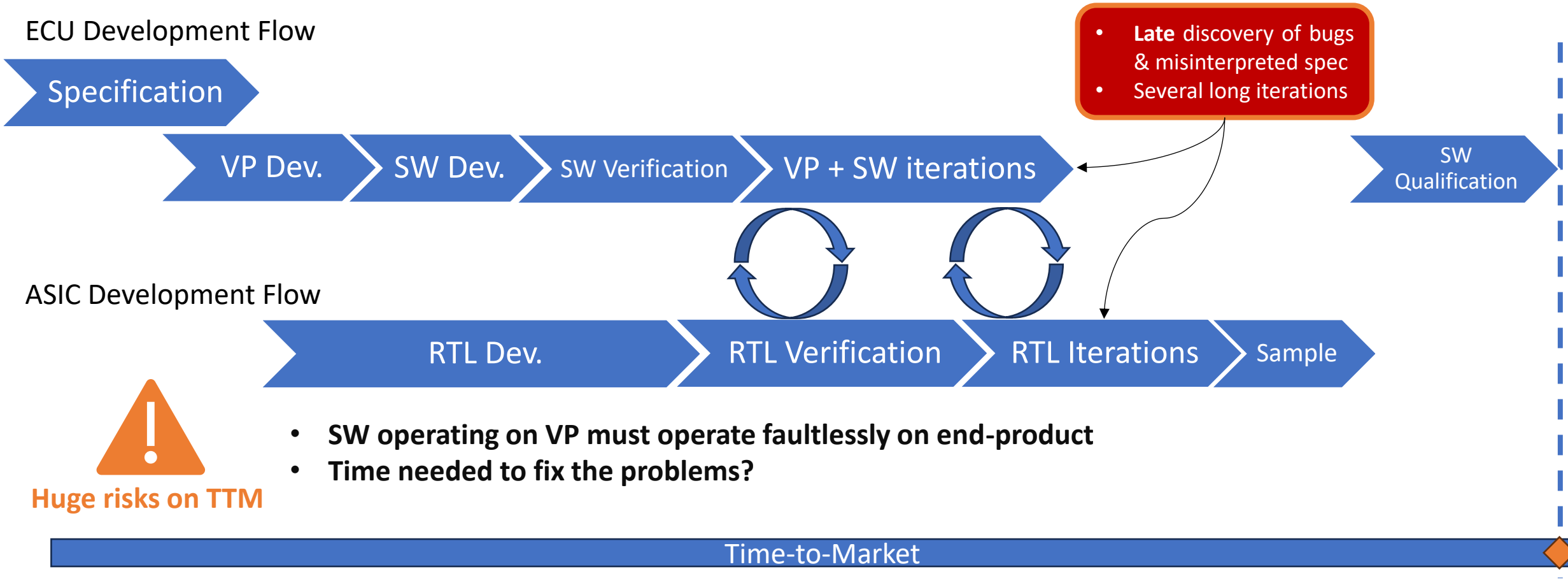
Javier Castillo, Mohammad Badawi, Jan-Hendrik Oetjens

TRISTAN RISC-V

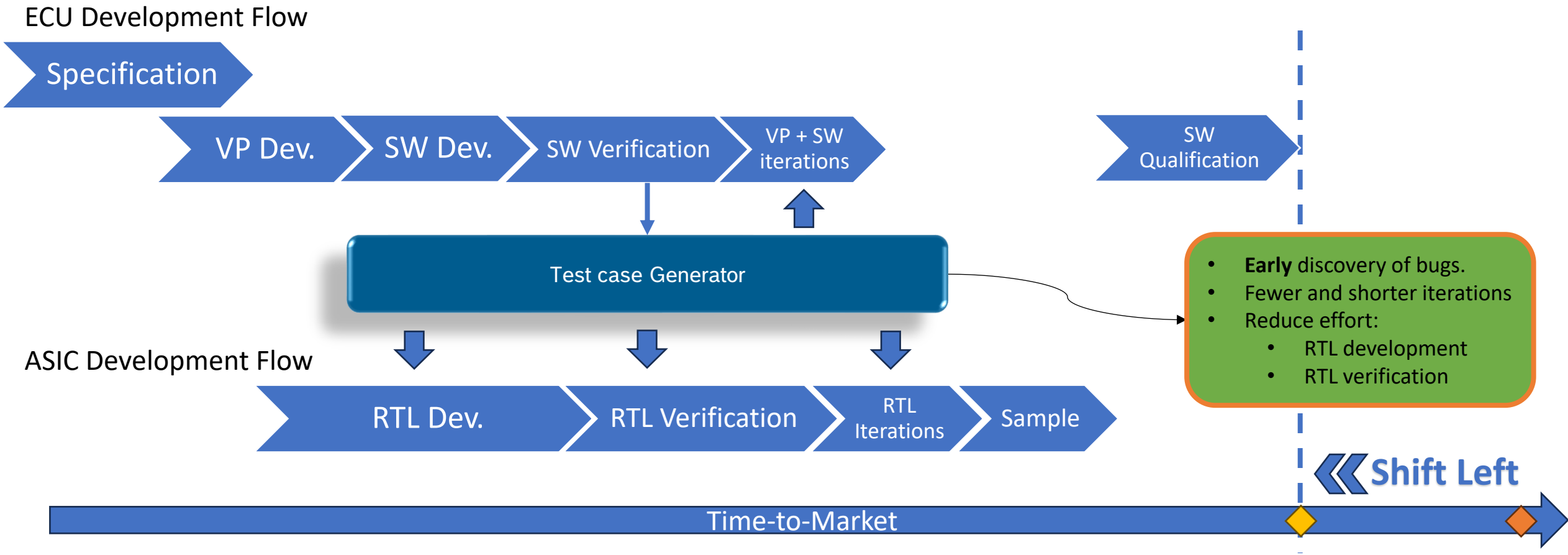
BOSCH

accellera SYSTEMS INITIATIVE

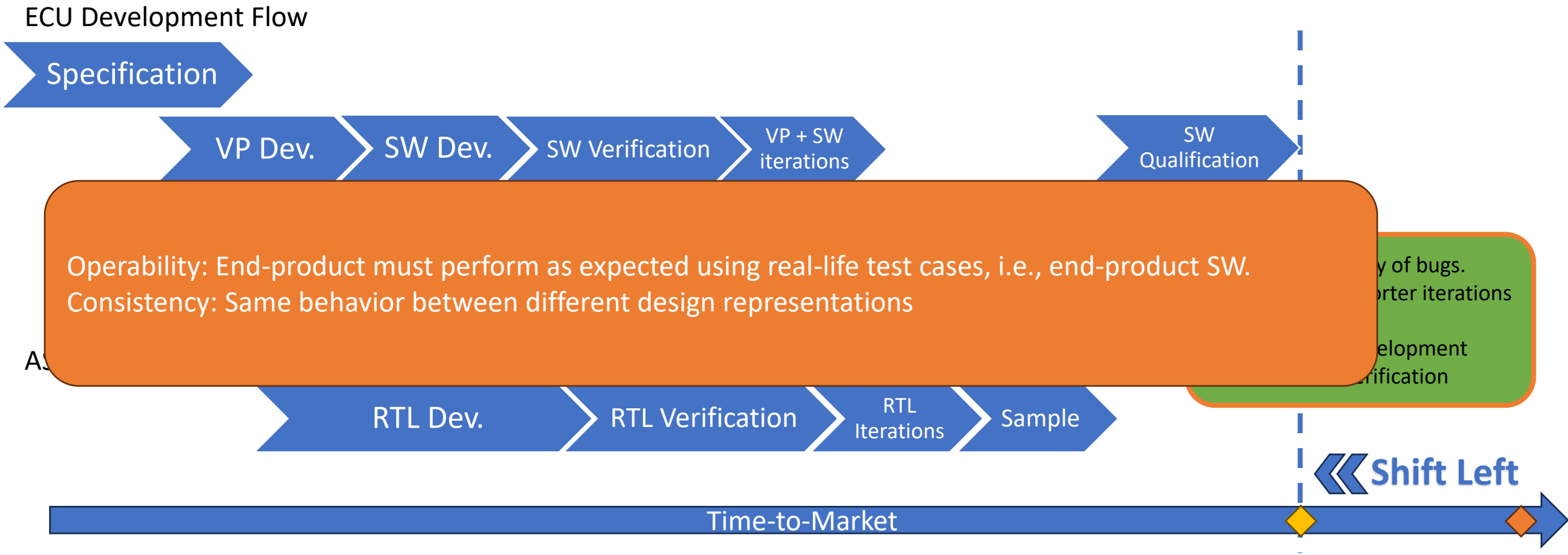
Development Flow: Individual Verification



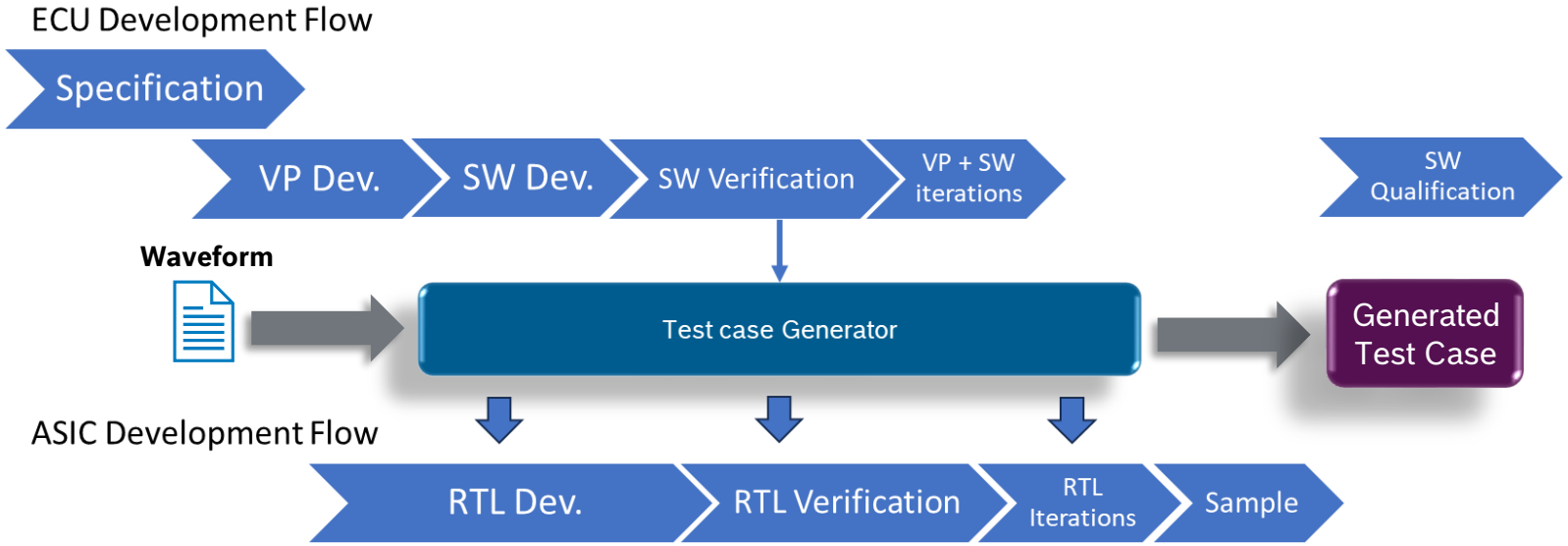
Development Flow: Cross-level Verification



Development Flow: Cross-level Verification

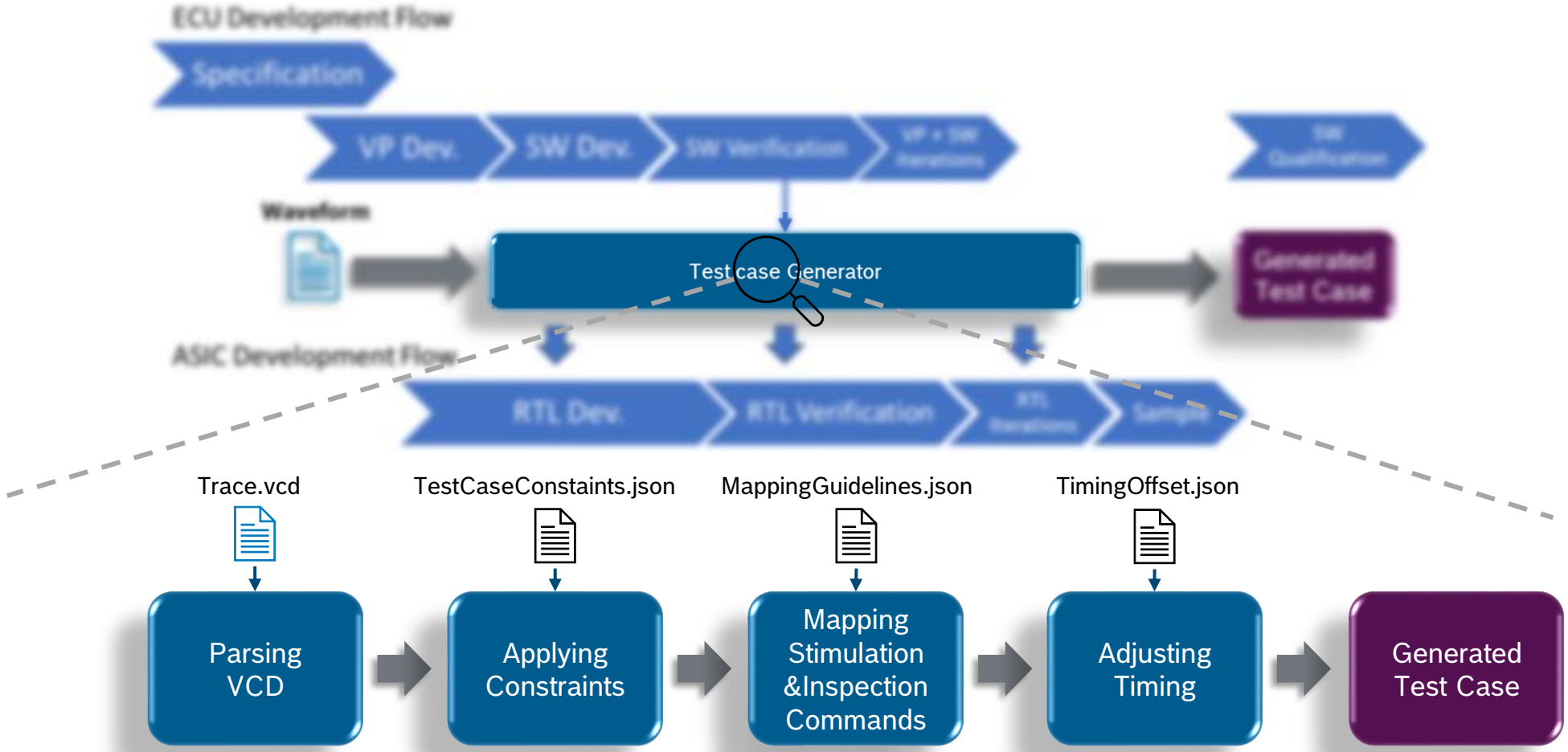


Approach: Automating Cross-Level Verification



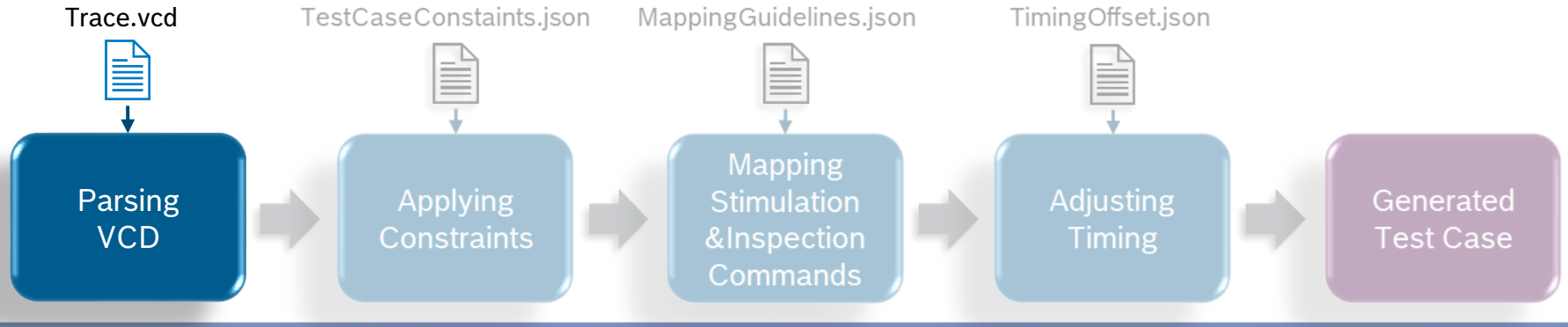
- Higher to lower abstraction (VP Waveform to RTL Test Case) → Refinement required
- Lower to higher abstraction (RTL Waveform to VP Test Case) → Ignore certain detail
- Overcome: Different expertise (SW, HW, FW) & confidentiality issues

Test Case Generator: Overview



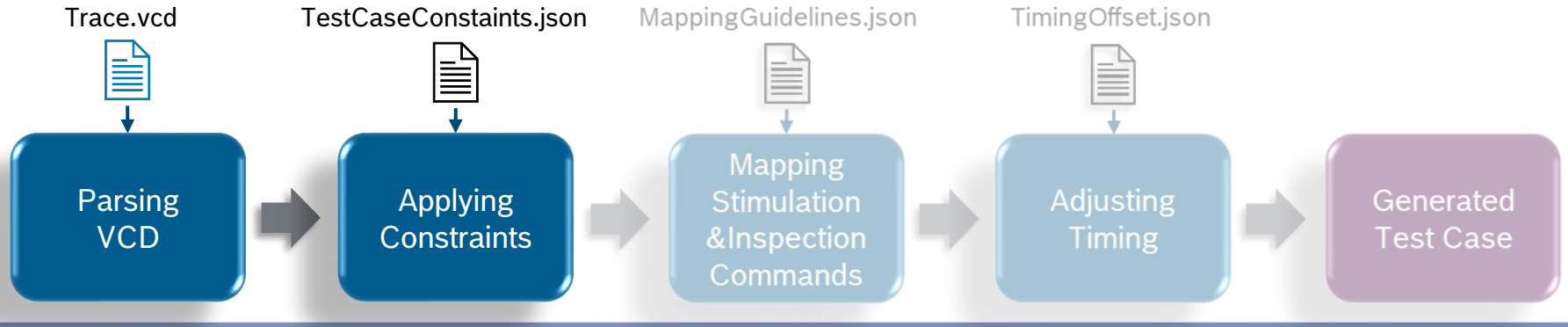
Test Case Generator: VCD Parsing

- Validate VCD
  - All configured signals should exist



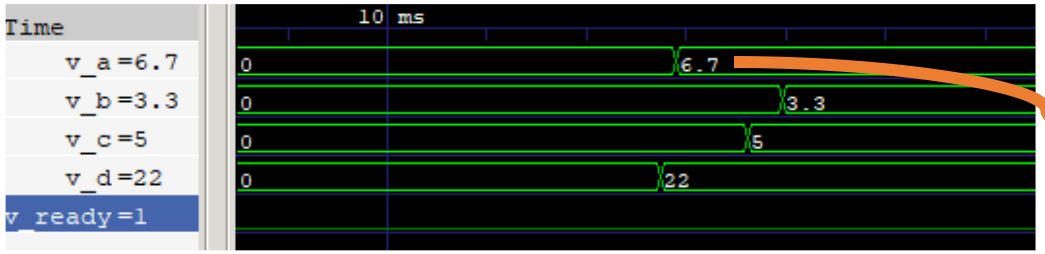
Test Case Generator: Timing Constraints

- Ensure timing followed for certain signals.
  - E.g., start, end and duration of TLM transactions in VP vs bus protocols in RTL
  - Prevent overlapping of transactions



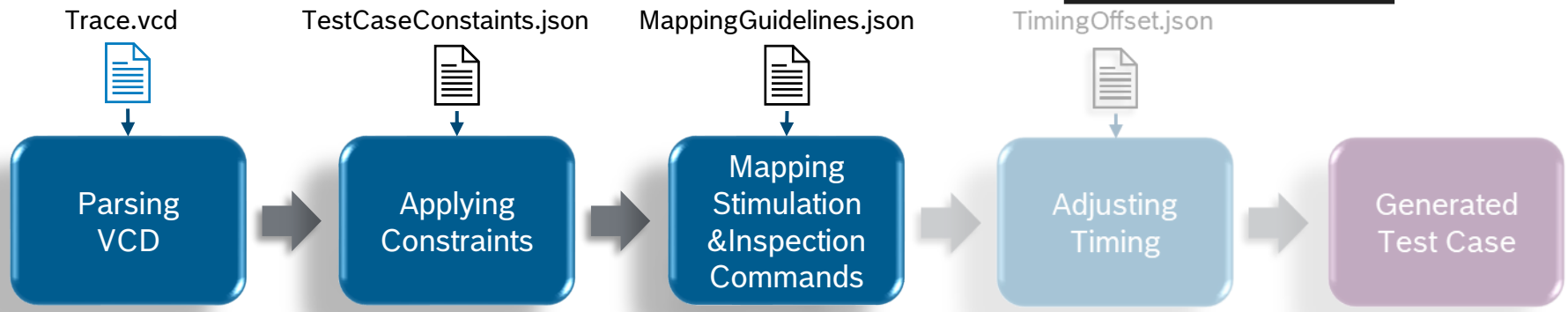
Test Case Generator: Mapping Signal Activity

- Dictates what code to generate in the test case file




```
#100us // time=12.50ms
CHECK(Vd === 22);
#240us // time=12.74ms
CHECK(Va === 6.7);
#730us // time=13.47ms
CHECK(Vc === 5);
#200us // time=13.67ms
CHECK(Vb === 3.3);
```

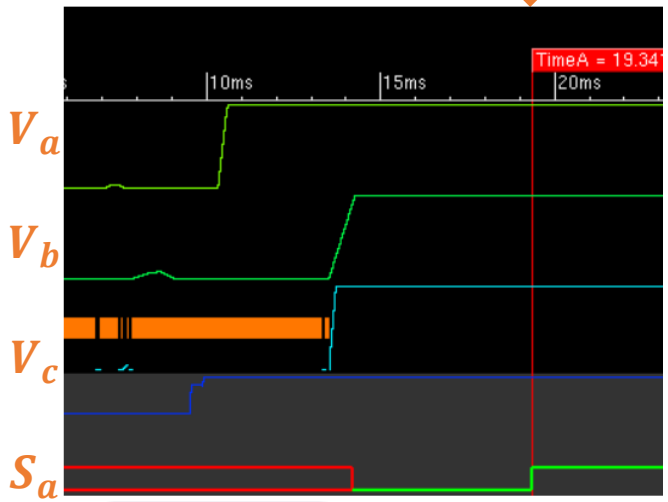
Composed of a set of stimulation/inspection commands

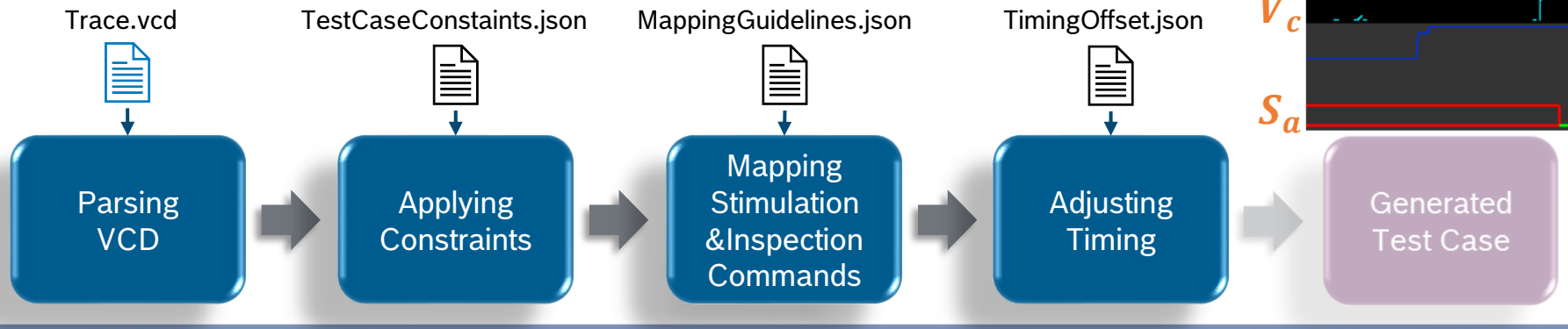


Test Case Generator: Time Adjustment

- Abstraction causes differences in timing
- Flexibility required:
  - Stimulation/inspection commands issued at the right time
  - Time-based adjustment
  - Event-based adjustment (circular dependency)

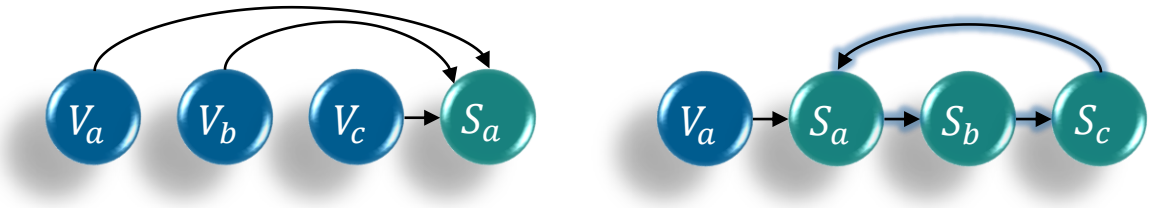
Check voltages on rising edge S_a

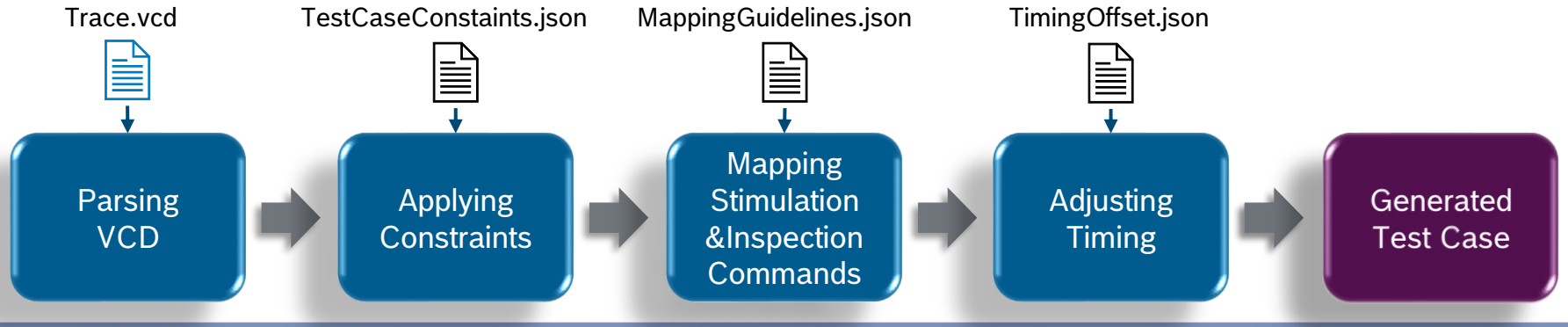




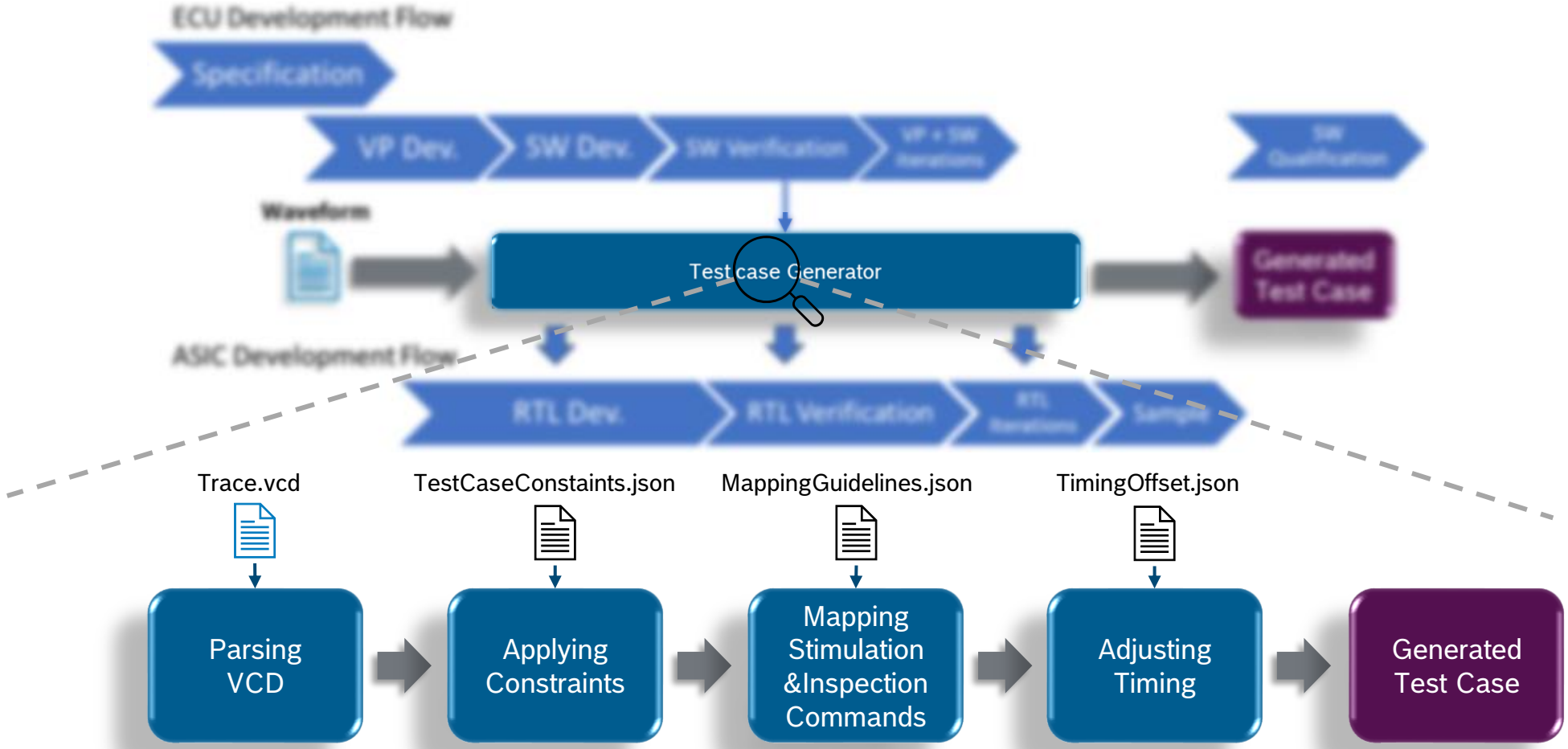
Test Case Generator: Resolving Circular Dependency

- Ensure a stable solution and successful generation of a test case
- Detect and handle circular dependency
  - Directed graph
  - Depth First Search Algorithm
  - User-specified number of iterations

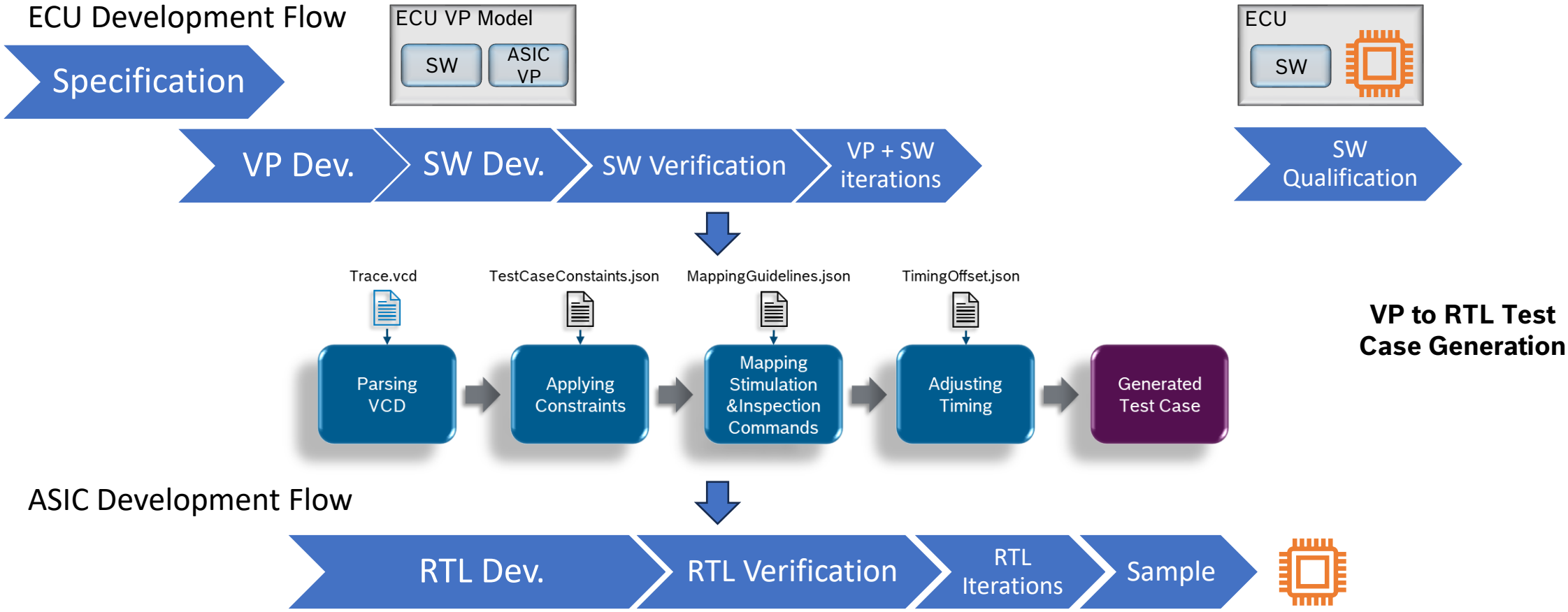




Test Case Generator: Complete Steps



Case Study: ASIC for Automotive Application



VP to RTL Test Case Generation

Results: Cross-Level Verification

Property	Automatic Test Case Generation	Manual Test Case Generation
Quality: consistency & operability	☑ Ensured at early stage	☒ Only possible at later stages
Bug detection	☑ Discovered early	☒ Would have been discovered late
Misinterpreted spec.	☑ Resolved early	☒ Discovered late
Dev. iterations	☑ Fewer and shorter	☒ More and longer
Collaboration between teams	☑ Expertise is not a barrier	☒ Developers from HW, SW and FW required
Effort	☑ Tests generated in seconds	☒ Manual test generation demands days/weeks
Risks and Costs	☑ Low	☒ High

Summary

- Cross-level verification flow
  - Bridges the gaps during ECU and ASIC development
  - Enhances collaboration without the need for expertise (SW, HW, FW)
- Real automotive application was used as case study
- Improved quality of spec. and design representations (VP and RTL)
- Ensured operability & consistency
  - End-product test cases verified on all design representations
- Reduced efforts, risks and costs

Questions

Acknowledgement

The TRISTAN project, nr. 101095947 is supported by Chips Joint Undertaking (CHIPS-JU) and its members Austria, Belgium, Bulgaria, Croatia, Cyprus, Czechia, Germany, Denmark, Estonia, Greece, Spain, Finland, France, Hungary, Ireland, Israel, Iceland, Italy, Lithuania, Luxembourg, Latvia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, Turkey and including top-up funding by Federal Ministry of Education and Research, BMBF (Germany).

accellera
SYSTEMS INITIATIVE