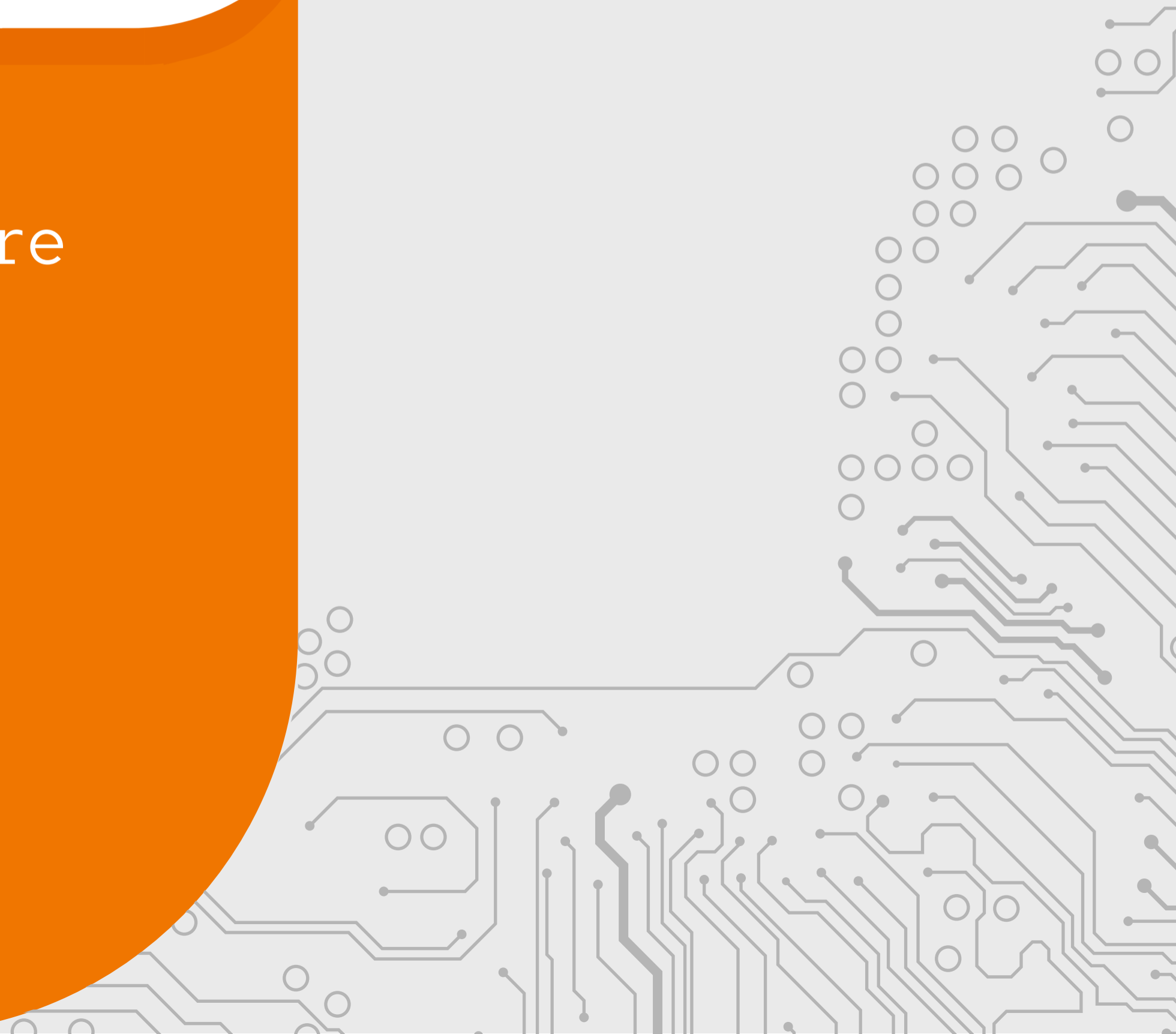


ecu.test

Produktbroschüre

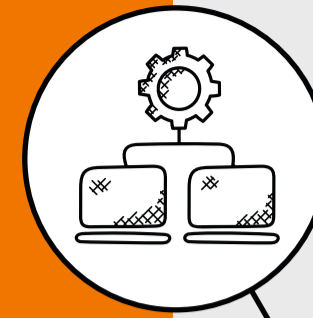


SOFTWARE TESTING. SIMPLIFIED.

Softwareentwicklung ist ein komplexer Prozess, bei dem mit jedem neuen Feature auch viele Testschritte verbunden sind. Je vielschichtiger die Software, umso umfangreicher werden die dafür benötigten Tests.

ecu.test ist unsere Software für komplexe Funktions- und Systemtests von Steuergerätesoftware. Mit dem Tool werden Testfälle für Automotive Software in allen Entwicklungsphasen bis hin zum fertigen Fahrzeug erstellt und automatisiert in unterschiedlichen Testumgebungen ausgeführt.

Unsere passgenauen Lösungen skalieren dabei dynamisch mit dem Testvolumen – von Einzelplatzsystemen bis hin zu parallelen Ausführungen auf verteilten Testsystemen in Clouds und Clustern.



ecu.test

Leistungsumfang

Testfallausführung in allen Entwicklungsstadien von Fahrzeugsoftware

Integrationsplattform für den flexiblen Zugriff auf verschiedene Testwerkzeuge und Testumgebungen (MiL/SiL/HiL/ViL)

Orchestrierung bestehender Toolketten

Editor zum einfachen Erstellen von Testfällen über Toolgrenzen hinweg

Vollständige Automatisierung der gesamten Testumgebung, auch auf skalierenden Systemen

Testausführung unter Windows und Linux, Desktop oder Container (u. a. Docker)

Anbindung an ALM-Systeme

Teamübergreifende Zusammenarbeit durch Diff- und SCM-Integration (Git)

Synchronisation, Analyse und Darstellung von Signalaufzeichnungen aus unterschiedlichen Quellen

SO FUNKTIONIERT DAS TESTEN MIT **ecu.test**

Alles beginnt mit einer Anforderung an eine neue, innovative Fahrfunktion und endet mit deren Freigabe nach erfolgreichem Test.

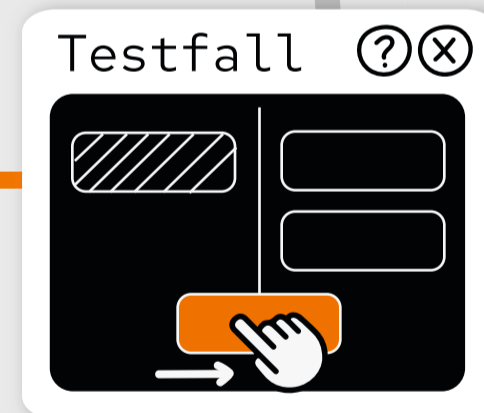
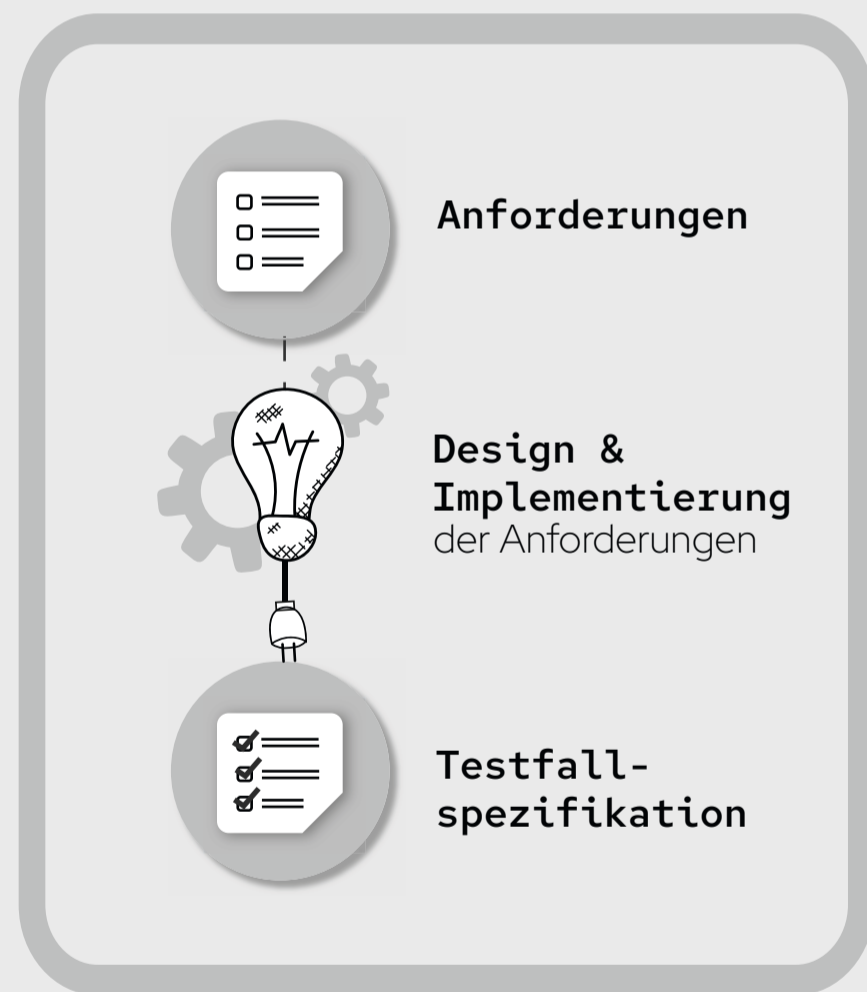
- Erstellen von Testfällen in der GUI per Drag&Drop
- Kombination multipler Signalquellen
- Testausführung mit verschiedenen Tools in allen Testdomains (ADAS/AD, Infotainment, Powertrain etc.)
- Erstellung von Diagnosen (ohne 3rd-Party-Tools)
- Durchführung von Kalibrierungen (ohne 3rd-Party-Tools)

Automatische Analysen von Traces und Erstellung von individuellen Reports

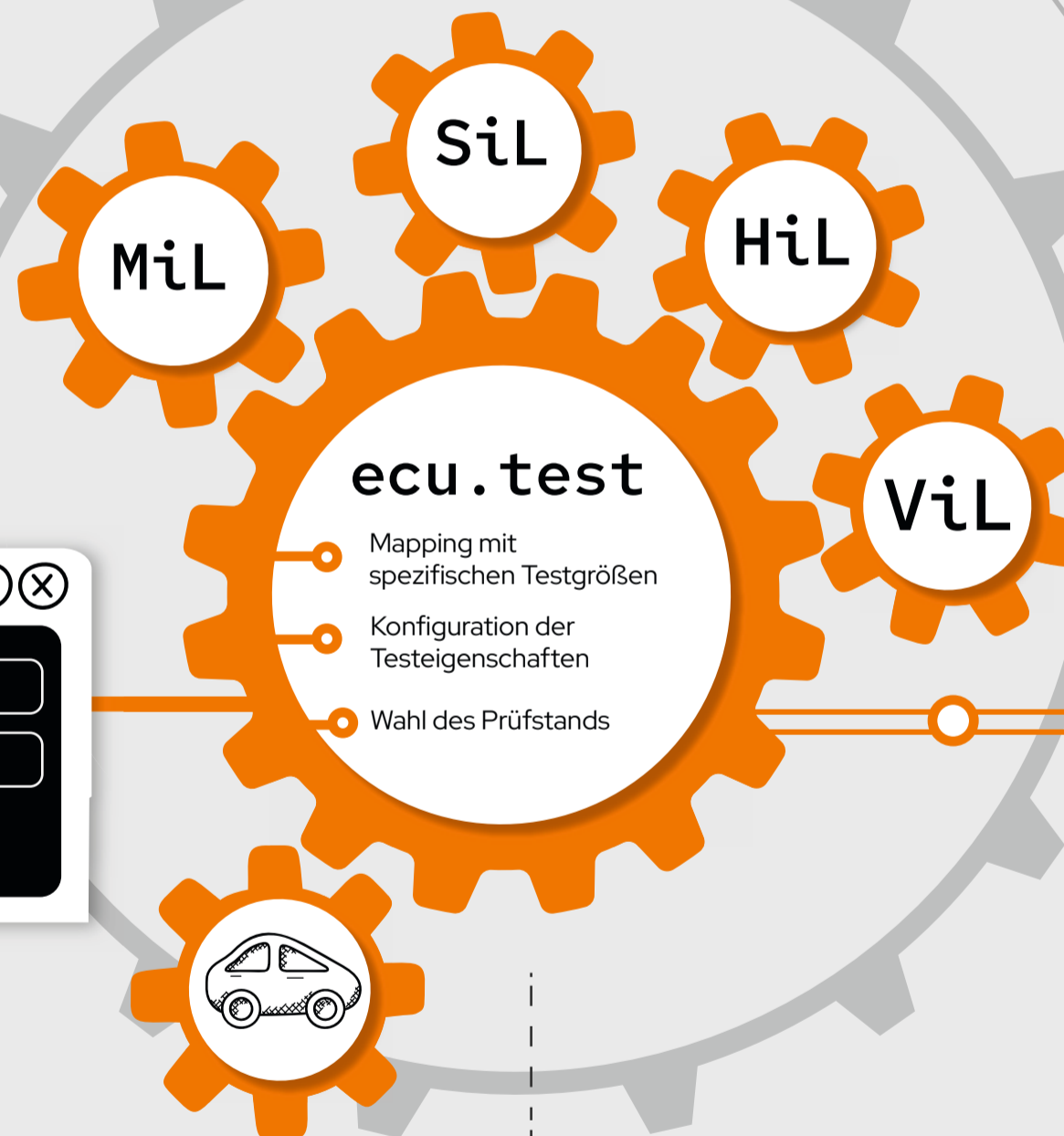
Testfälle gemeinsam nutzen und wiederverwenden

Individuelle ecu.test-Anpassungen (API, Python)

Anbindung an ALM-Systeme



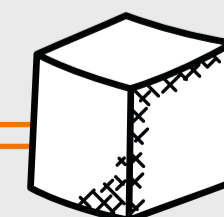
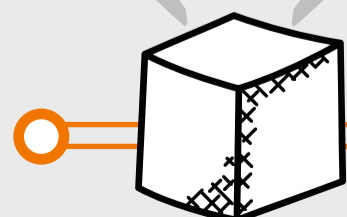
Testfall
Technische Anleitung zur Testausführung



Testausführung
Testdurchführung in gewählter Testebene und Testumgebung

Review der Testergebnisse
Report mit allen Daten der Testausführung

Export der Testergebnisse



DEINE VORTEILE AUF EINEN BLICK

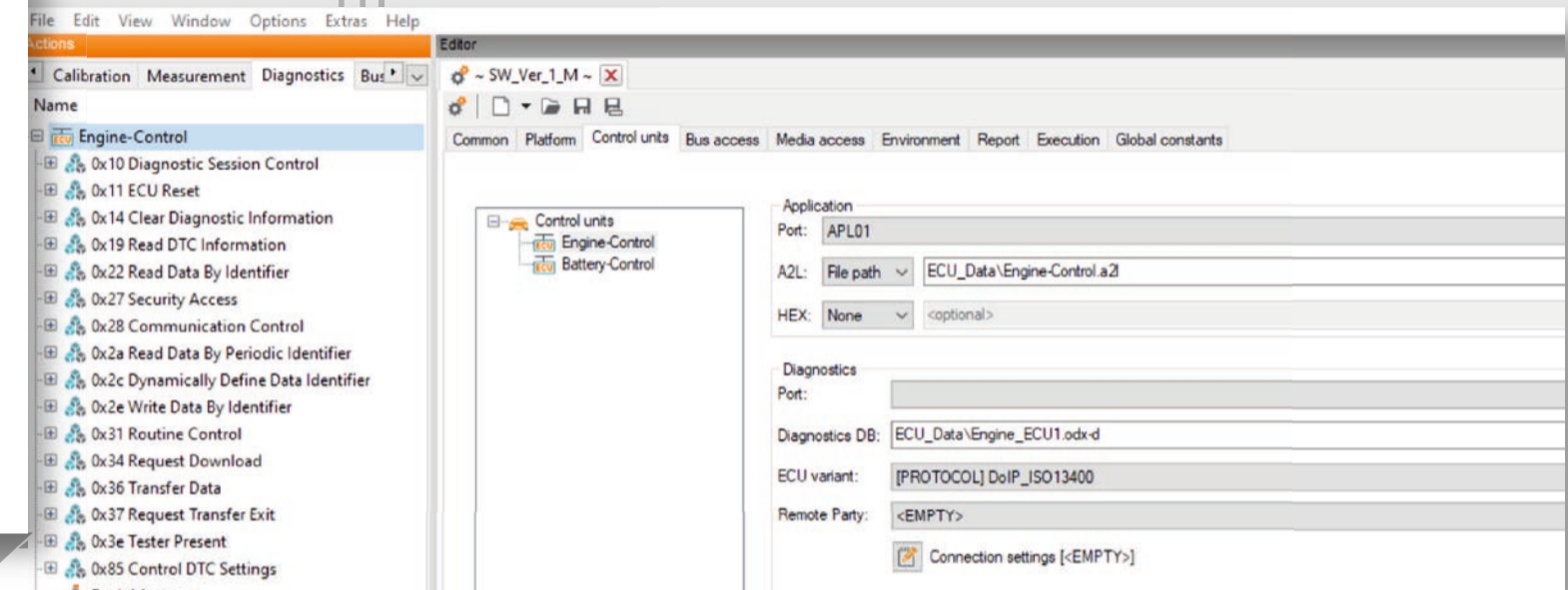
ecu.test

meistert die Testautomatisierung

1

Spricht fließend Automotive

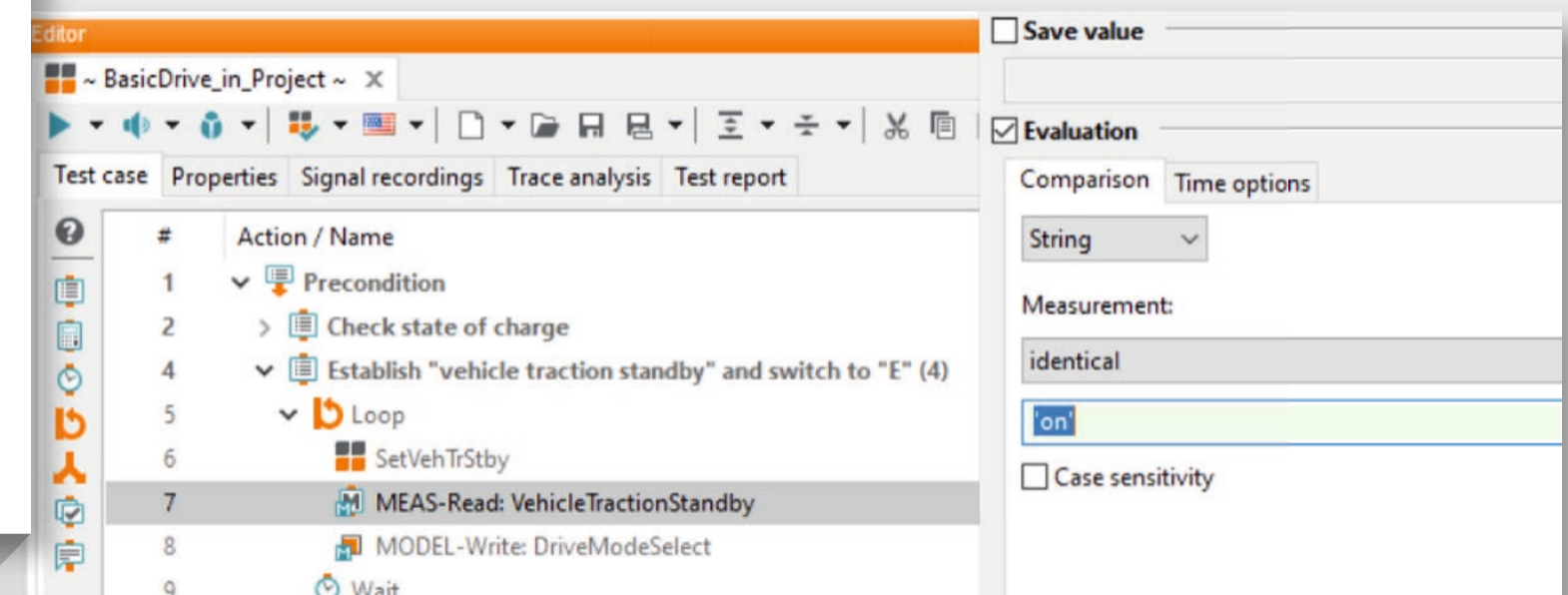
Ob ARXML, A2L, DBC oder ODX – ecu.test verarbeitet Daten jedes gängigen Automotive-Formats. Auch der Zugriff auf Bild- und Audiodaten der Headunit, auf Sensordaten von ADAS-Systemen oder auf die Touchscreen-Steuerung von Handhelds ist möglich.



2

Grafisch geführte Testfallerstellung

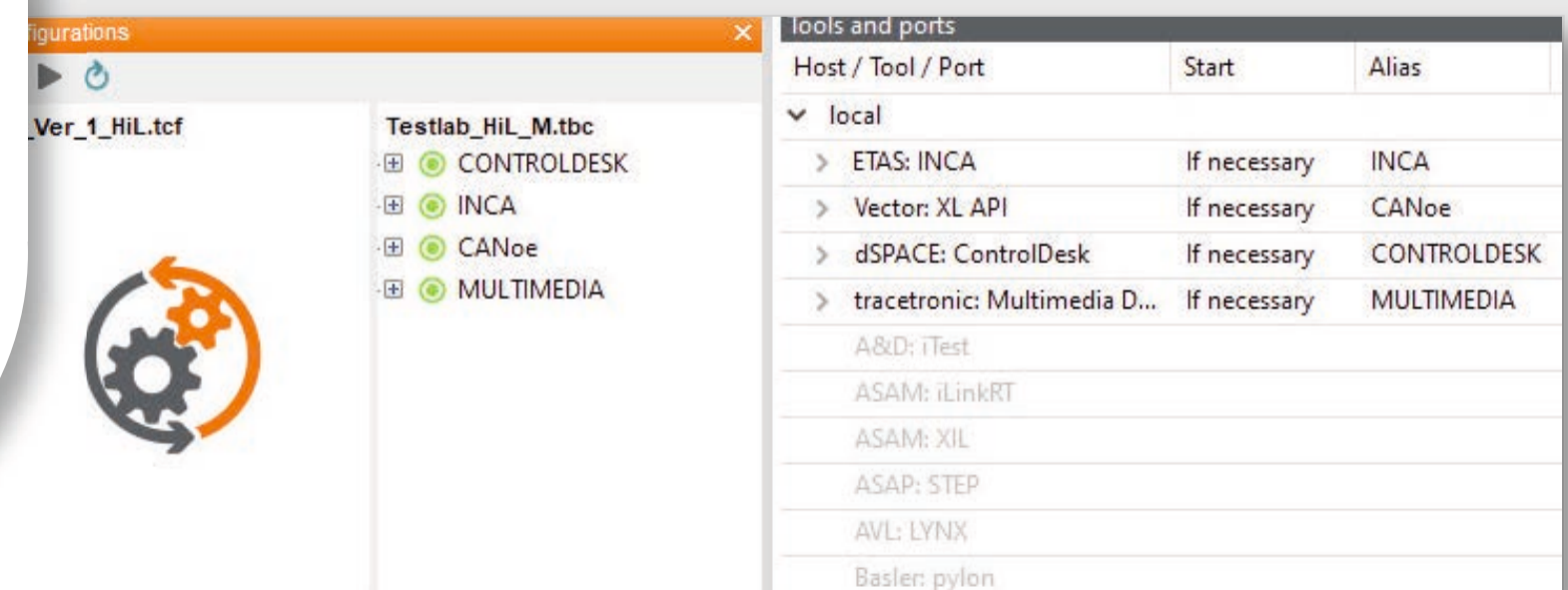
Einmal erstellte Testfälle sind aufgrund der generischen Testbeschreibung weitestgehend unabhängig von spezifischer Hard- und Software der Prüfumgebungen. Dadurch lassen sich Tests vielseitig und wiederholt in verschiedenen Teststufen (MiL, SiL, HiL, ViL) und Testbereichen (Powertrain, Infotainment oder ADAS) einsetzen.



3

Verbindet Tools verschiedener Hersteller

Über standardisierte Zugriffe und vollständig integrierte Toolschnittstellen lassen sich mehr als 80 Tools anbinden. Dabei wird die Testfall-Tool-Kommunikation vollständig abstrahiert, um den unveränderten Testfall auch mit verschiedenen Tools ausführen zu können. Auch deine Inhouse-Lösungen kann ecu.test automatisieren.



DEINE VORTEILE AUF EINEN BLICK

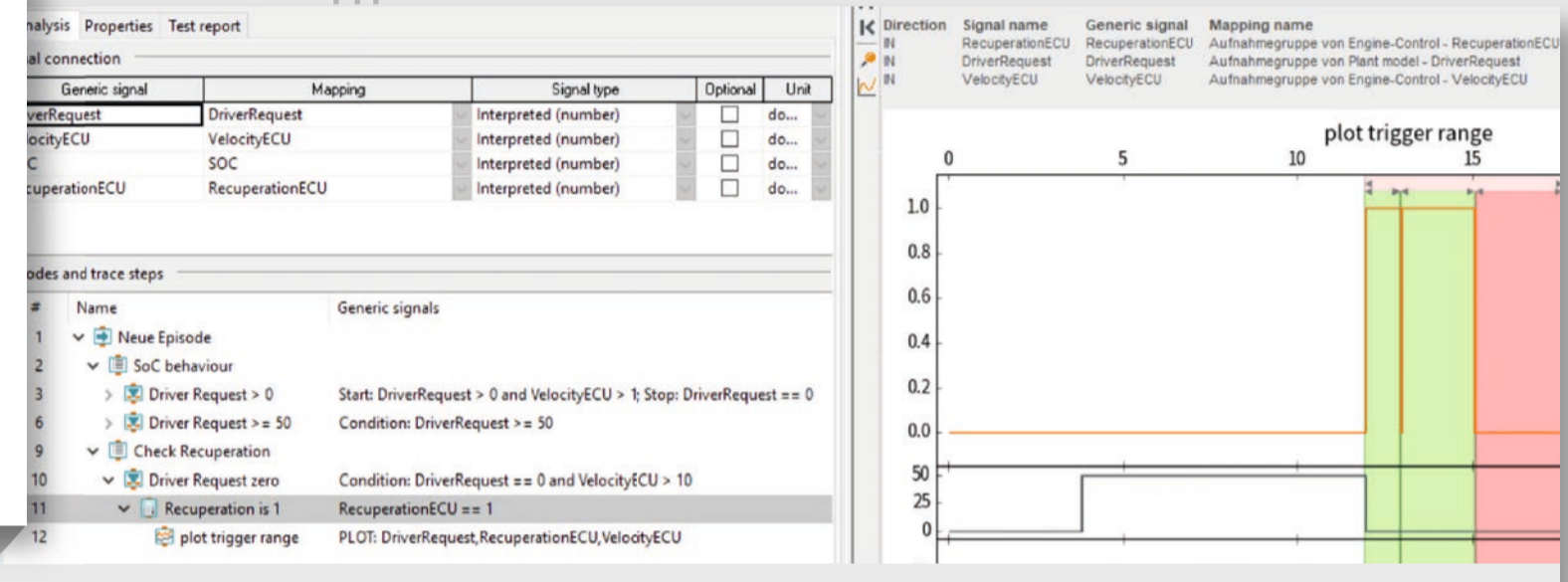
ecu.test

meistert die Testautomatisierung

4

Performante Signalanalyse

Für alle E/E-Komponenten und Systeme im Fahrzeug können die Signalverläufe über die Dauer eines Testfalls aufgezeichnet und ausgewertet werden. Wenn dabei Signale aus unterschiedlichen Quellen stammen, können sie automatisch synchronisiert werden.



5

Generiert strukturierte Ergebnisse

Während der Testausführung werden alle relevanten Daten und auftretenden Werte gespeichert. Sie bilden die Grundlage für die automatische Erstellung von Testreports, die vollumfänglich sind und Details zu jedem Testschritt enthalten.

#	Action/Name	Value	Expected value	Comment	Evaluation	Test time [s]
1	Precondition					0.014
12	Action				FAILED	3.802
13	Set driver request to 100 %				FAILED	3.802
15	Check velocity				FAILED	3.806
16	Multi-Check	True for all test steps (Wait until true: 270 trials ...)	All TS		FAILED	3.807
17	MODEL-Read: VelocityModel	47.21	value > 50	Read the vehicle...	FAILED	3.807
18	BUS-Read: VelocityBus	47.268	value > 50	Read the vehicle...	FAILED	3.807
19	MEAS-Read: VelocityECU	47.29	value > 50	Read the vehicle...	FAILED	3.807
20	Set driver request to 0 %					7.816
22	Check recuperation				SUCCESS	7.819

6

Ist vielseitig erweiterbar

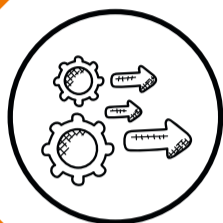
1. In ecu.test ist alles über API anbindbar.
2. Das Tool ist mit Python erweiterbar.
Für eigene Lösungen und Erweiterungen ist das ideal. Denn so lässt sich ecu.test schnell und einfach auf deine Bedürfnisse zuschneiden, sowohl bei der Testfallentwicklung als auch bei der Traceanalyse.

```
def ReadImage(self) -> tuple[NDArray[Any], NDArray[Any] | None]:  
    """  
    Reads an image from e.g. an external source and makes it available  
    be implemented for a port of type 'IMAGE'.  
    """  
    return self._cam.GetFrame(), None  
  
def GetTouchInput(self) -> ExampleTouchHandler:  
    """  
    Returns an object that provides the interface for processing touch  
    this method we indicate to ecu.test that this port supports touch in  
    be omitted if the port does not support touch input.  
    """  
    return ExampleTouchHandler(self._cam)
```

10 GRÜNDE FÜR ecu.test

ecu.test

ist immer eine gute Idee



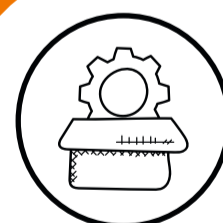
Testfälle sind jederzeit und überall automatisiert ausführbar, denn sie sind so geschrieben, dass sie in allen Testsystemen und Testdomänen einsetzbar sind.



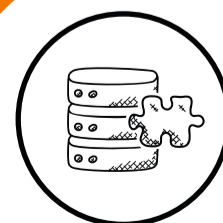
Weltweit verteilte Teams können einfach zusammenarbeiten, da Workspaces, Packages und Bibliotheken gemeinsam genutzt und wiederverwendet werden können.



Testdurchsatzraten können erheblich gesteigert werden, da zeitaufwendige Analysen außerhalb der teuren Prüfstände durchführbar sind.

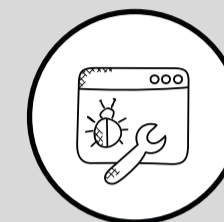


Testergebnisse sind in anderen Systemen weiterverwendbar, da sie in alle Formate konvertierbar sind.

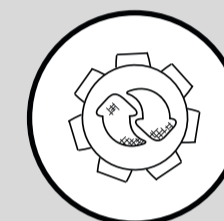


Daten sind jederzeit verfügbar und für Teams leicht austauschbar, da sie im XML-Format gespeichert und in Git/SVN verwaltet werden.

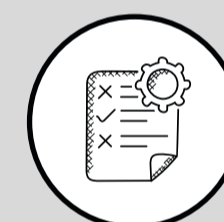
Softwarefehler werden schnell gefunden, da Tests reproduzierbar sind und verlässliche Ergebnisse liefern.



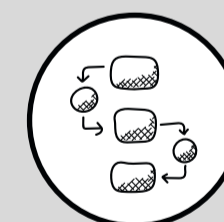
Testfälle können iterativ entwickelt und jederzeit ausgeführt werden, da immer eine direkte Verbindung zum Testsystem besteht.



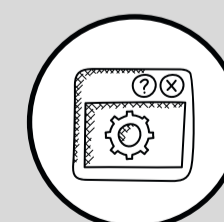
Einsteiger können sehr einfach Testfälle erstellen, da sowohl die Bedienung der GUI als auch der Zugriff auf die darunter liegenden Tools unkompliziert ist.

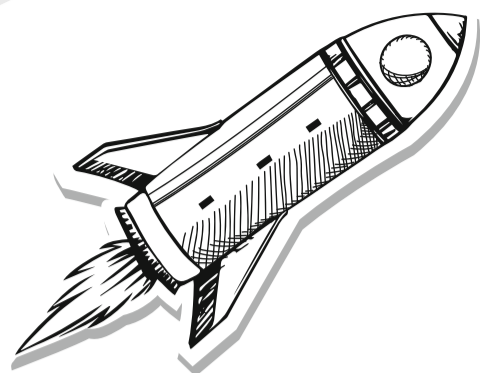


Testschritte sind detailliert nachvollziehbar, da die Struktur der automatisch erstellten Reports mit der Testfallstruktur übereinstimmt.



Testen in heterogenen Testlandschaften ist uneingeschränkt möglich, da sich weitere (Inhouse-) Tools über standardisierte API-Schnittstellen und Python-Skripte nahtlos in bestehende Systeme integrieren lassen.





RELEASES

Wir releasen vier Softwareversionen pro Jahr und informieren alle Anwenderinnen und Anwender umfangreich über neueste Features, Verbesserungen sowie An- und Abkündigungen.

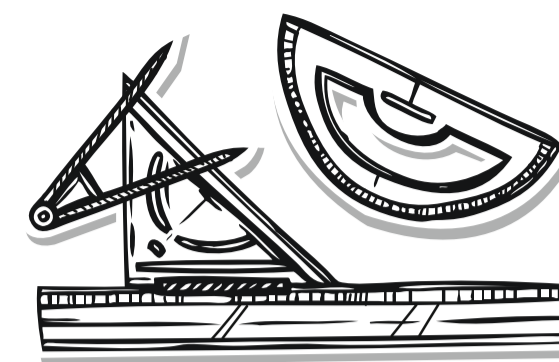
Im Sinne inkrementeller agiler Entwicklung stellen wir zahlreiche Vorabversionen zur Verfügung.



SUPPORT

Von all unseren Standorten bieten wir umfassende technische und strategische Unterstützung:

- Ersteinrichtung
- Allgemeine Nutzung der Software
- Problemanalysen und -behebung
- (Best practises) Beratung
- Testsystementwicklung
- Anpassung der Testmethoden und Teststrategien auf deine Workflows



ENGINEERING

Als Anbieter für End-to-End-Lösungen entwickeln wir durchgehende Workflows, um Softwaretests sehr detailliert, übergreifend und auf einem sehr hohen Automatisierungsgrad durchzuführen.

Auch auftragspezifische Optimierungen an bestehenden Systemen sind möglich – von einfachen Testfällen bis hin zu komplexen, dynamisch skalierten Automatisierungslösungen in der Cloud.

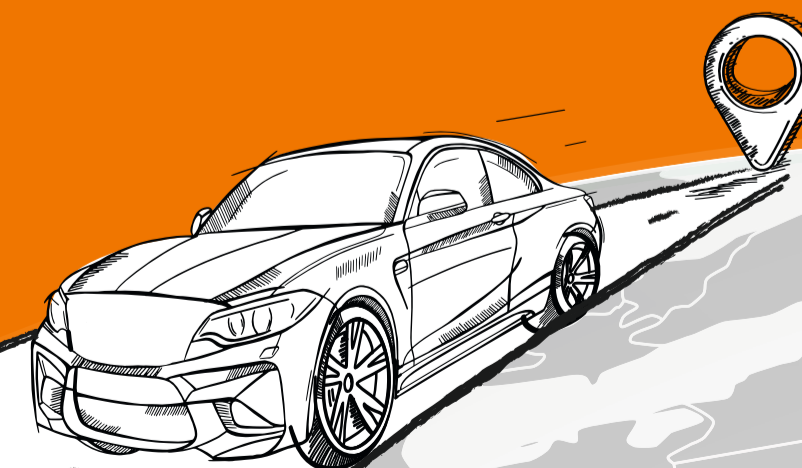


tracetrone unterstützt Hersteller und Zulieferer der Automobilindustrie bei der Entwicklung hochkomplexer Fahrzeugsoftware mit Softwareprodukten und maßgeschneiderten Dienstleistungen. Der Fokus liegt auf Lösungen für einen automatisierten Testprozess in allen Phasen der Softwareentwicklung – von Komponententests bis hin zu Integrationstests im Fahrzeug.

tracetrone wurde im Jahr 2004 als Uni-Start-Up an der TU Dresden gegründet und hat sich seitdem zu einem globalen Unternehmen entwickelt. Mittlerweile sind über 400 spezialisierte Mitarbeitende, Studierende und Auszubildende beschäftigt.

Die tracetrone Gruppe hat ihren Hauptsitz in Dresden und weitere Standorte in München, Ingolstadt, Stuttgart, Hamburg sowie in den USA, Südkorea, Japan und China.

In weltweit über 30 Ländern sind unsere Produkte und Lösungen bei mehr als 400 Kunden, wie Audi, BMW Group, BYD, Continental, Daimler, Ford, John Deere, Magna, Mercedes, Porsche, Rivian, Siemens, Stellantis, Valeo oder Volkswagen im Einsatz.



www.tracetrone.de