

# Automotive Day - München

## PreConference

Tag 1 - 05.11.2024

9.30 - 13.00 Uhr

### Durchgängige Toolgestützte Entwicklung eines Automotive Systems

Präsentiert von: Hartmut Wittke (BTCes), Jan Richter (Siemens), Michael Spieker (SodiusWillert)

#### Einführung:

In dieser Session demonstrieren Sodius-Willert, BTC Embedded Systems und Siemens gemeinsam exemplarisch, live, und Tool-basiert die durchgängige Entwicklung eines adaptiven Abstandstempomats von Anforderungen bis zur integrierten Anwendung im AUTOSAR-Stack. Hierbei werden die unterschiedlichen Entwicklungsschritte der einzelnen Engineering-Disziplinen im jeweiligen Tool vorgestellt.

Ein besonderer Fokus liegt auf effizienter Zusammenarbeit der jeweiligen Anwendergruppen über Toolgrenzen hinweg, ermöglicht durch Toolintegrationen und Automationen, auf Risikoreduktion durch frühes und durchgängiges Testen, sowie auf erleichterte Compliance mittels Ende-zu-Ende Traceability von Anforderungen durch die involvierten Modellebenen hindurch bis zum Quellcode.

#### Inhalte

In dieser Session demonstrieren Sodius-Willert, BTC Embedded Systems und Siemens gemeinsam exemplarisch, live, und Tool-basiert die durchgängige Entwicklung eines adaptiven Abstandstempomats von Anforderungen bis zur integrierten Anwendung im AUTOSAR-Stack. Hierbei werden die unterschiedlichen Entwicklungsschritte der einzelnen Engineering-Disziplinen im jeweiligen Tool vorgestellt.

Ein besonderer Fokus liegt auf effizienter Zusammenarbeit der jeweiligen Anwendergruppen über Toolgrenzen hinweg, ermöglicht durch Toolintegrationen und Automationen, auf Risikoreduktion durch frühes und durchgängiges Testen, sowie auf erleichterte Compliance mittels Ende-zu-Ende Traceability von Anforderungen durch die involvierten Modellebenen hindurch bis zum Quellcode.

Wir beginnen als Systemarchitekten mit Systemanforderungen und ihrer Erfassung, Analyse, und Überführung in Systemelemente anhand der SysML-Sprache in IBM Rhapsody, sowie deren Validierung und frühe Tests mittels TextConductor. Davon ausgehend zeigen wir die effiziente Zusammenarbeit von System- und Softwarearchitekten, indem wir die initiale Softwarearchitektur mittels Modell-zu-Modell-Transformation automatisiert und schnell wiederholbar aus dem SysML-Modell ableiten. Der Softwarearchitekt verfeinert und ergänzt diese dann unter Verwendung spezialisierter Editoren für AUTOSAR-Aspekte zu einer vollständigen AUTOSAR-Softwarearchitektur mit Anteilen in der Adaptive und Classic Plattform und generiert daraus den Implementierungsvertrag in Form einer RTE-API. All diese Schritte zeigen wir in IBM Rhapsody. Das Softwareentwicklungsteam entwickelt das Verhalten der Softwarekomponenten auf Basis des RTE-Vertrags modellbasiert mittels UML in IBM Rhapsody, oder in externen Programmierumgebungen. Die Codegenerierung und Softwareunittests mit TestConductor führen schließlich zu qualitativ hochwertigem Produktivcode in C und C++.

Als nächstes zeigen wir die Entwicklung der Elektrik/Elektronik (E/E)-Plattform, welche Softwarefunktionen trägt und ausführt. Der E/E-Architekt importiert die relevanten Funktionen aus

SysML in Rhapsody in Capital Systems Architect, wo er die E/E-Plattformtopologie pflegt und erweitert, die Funktionen auf Komponenten abbildet und verschiedene Allokationsszenarien vergleicht. Der Netzwerk- und Busexperte optimiert anschließend das Routing der Signale und ihre Paketierung für optimale Bandbreitenausnutzung und garantiertes Einhalten der Zeitanforderungen für jedes einzelne Signal, in Capital Network Designer. Hieraus entsteht für jedes Steuergerät ein ECU-Extrakt, das zusammen mit den fertigen Softwarekomponenten den Input für die Steuergeräteintegration darstellt.

Der Steuergeräteintegrator konfiguriert auf dieser Basis die AUTOSAR Basissoftwaremodule bzw. Middleware in Capital Embedded Integrator, um den Anforderungen an Kommunikation, Datenpersistenz, IO-Interaktionen, funktionale Sicherheit, Cybersicherheit, und Diagnose gerecht zu werden. Virtuelle Steuergerätehardware für gängige Hardwareplattformen erlaubt Integrationstests mit hoher Aussagekraft in Capital Embedded Virtualizer, noch bevor ECU-Hardwaremuster verfügbar sind.

Durch die enge Toolintegration kann TestConductor/Rhapsody Integrationstests in Capital Embedded Virtualizer ausführen und die Ergebnisse dokumentieren, sowohl interaktiv durch das Softwareentwicklungsteam, als auch automatisiert in einer CI/CD Pipeline. Durch die durchgängige Testunterstützung vom multi-Domänen Systemmodell in SysML bis zum integrierten Anwendungscode ist TestConductor das Tool der Wahl für Feature-Tester und Software-Tester, von der Softwareunit bis hin zum integrierten Feature.

#### **Kontakt Daten:**

Sigrun Willert  
[swillert@sodiuswillert.com](mailto:swillert@sodiuswillert.com)