

Inhaltsverzeichnis

1 Stand der Technik bei der Software-Entwicklung im Automobil	1
1 Einleitung	3
1.1 Trends für die Software-Entwicklung im Automobil	3
1.2 Aktuelle Herausforderungen bei der Absicherung von Fahrzeug-Software	5
1.3 Software-Qualitätssicherung in frühen Entwicklungsphasen	7
1.4 Übersicht über die Arbeit	9
2 Software bei der Elektrik/Elektronik im Automobil	13
2.1 Bestandteile der Elektrik/Elektronik	13
2.1.1 Systeme und Vernetzung	14
2.1.2 Steuergeräte und Software	16
2.1.3 Funktionen	19
2.2 Elektrik/Elektronik-Entwicklung	19
2.2.1 Entwicklungsprozesse	20
2.2.2 Entwicklungsrollen	23
2.2.3 Entwicklungswerkzeuge	25
2.3 Modellbasierter Software-Entwurf	26
2.4 Zusammenfassung	28
3 Qualitätssicherung von Fahrzeug-Software	31
3.1 Einordnung der Software-Qualitätssicherung	31
3.2 Methoden zur Software-Prüfung	33
3.2.1 Statische Prüfung	34
3.2.2 Dynamische Prüfung	39
3.3 Testaktivitäten und Automatisierung	46
3.4 Testziele in der E/E-Entwicklung	48
3.5 Zusammenfassung	50
4 Weiterführende Ansätze zur frühen Sicherung der Software-Qualität	51
4.1 TITUS	51
4.1.1 TITUS-Software-Architektur	52
4.1.2 TITUS-Entwicklungsprozess	53
4.2 AUTOSAR	53
4.2.1 AUTOSAR-Software-Architektur	54

4.2.2	AUTOSAR-Methodologie	56
4.2.3	AUTOSAR-Beschreibungsformat und Werkzeuge	56
4.3	ATESST2	57
4.3.1	EAST-ADL2	57
4.3.2	ATESST2-Systemmodell	58
4.4	TIMMO	58
4.4.1	TIMMO-TADL	59
4.4.2	TIMMO-Methodologie	60
4.5	Gap-Analyse bisheriger Ansätze	60
4.6	Anforderungen und Randbedingungen	62
4.7	Zusammenfassung	65
II	Frühe Software-Qualitätssicherung mittels virtueller Integration	67
5	Virtuelle Integration basierend auf AUTOSAR	69
5.1	Zentrales Konzept der virtuellen Integration	69
5.2	Testziele der virtuellen Integration	72
5.3	Integrationssichten und -tiefen	75
6	Konzeption einer formalen AUTOSAR-Integrationsplattform	79
6.1	Analyse der AUTOSAR-Software-Architektur	79
6.2	Aufbau der Integrationsplattform	82
6.2.1	Applikation und RTE	83
6.2.2	Abstrakte Basis-Software	84
6.2.3	Stimulation und Auswertung	89
6.3	Ansätze zur Stimulation und Auswertung	92
6.3.1	Globale Variablen	93
6.3.2	RTE-Aufrufe	93
6.3.3	Virtuelle ECU-Schnittstelle	93
6.4	Sichten auf die Integrationsplattform	94
6.4.1	SWC-Integrationsplattform	94
6.4.2	System-Integrationsplattform	95
6.4.3	ECU-Integrationsplattform	96
6.5	Anwendungsfälle	97
6.5.1	Fehlereinstreuung	97
6.5.2	Timing	99
6.5.3	NV-Speicherung	102
6.5.4	Diagnose	104
7	Eingliederung der virtuellen Integration in die Software-Entwicklung	109
7.1	Entwicklungsprozess	109
7.1.1	Neue Stufen für eine virtuelle Integration	110
7.1.2	Zuordnung der Testszenarien auf die Integrationsstufen	111

7.1.3 Zeitlicher Ablauf einer virtuellen Integration	113
7.2 Werkzeugkette	114
7.3 Testadaptierung	116
7.3.1 Master-Client-Adaptierung	116
7.3.2 Master-Master-Adaptierung	117
8 Prototypische Umsetzung des Konzepts	119
8.1 Prozesstechnische Umsetzung	119
8.1.1 Virtuelle Systemintegration	120
8.1.2 Virtuelle ECU-Integration	121
8.1.3 Entwicklungsaktivitäten bei einer virtuellen Integration	124
8.2 Werkzeugtechnische Umsetzung	127
8.2.1 Werkzeuge für eine virtuelle Integration und Test	127
8.2.2 Anbindung der untersuchten Werkzeuge	132
III Evaluierung und Diskussion	135
9 Fallstudien und Konzeptevaluierung	137
9.1 Innenlichtsystem	137
9.1.1 Software-Struktur des Innenlichtsystems	137
9.1.2 Virtuelle Integration und Test des Innenlichtsystems	138
9.2 Body-Controller-Front (BC_F)	141
9.2.1 Software-Struktur des BC_F	142
9.2.2 Virtuelle Integration und Test des BC_F	143
9.3 Aufwand-/Nutzenanalyse	145
9.3.1 Aufwand für die Applizierung der virtuellen Integration	145
9.3.2 Nutzen der virtuellen Integrationstests	149
9.4 Schlussfolgerung	151
10 Zusammenfassung und Ausblick	153
10.1 Zusammenfassung	153
10.2 Ausblick	154
Anhang	157
Abbildungsverzeichnis	159
Tabellenverzeichnis	161
Abkürzungsverzeichnis	163
Glossar	165
Literaturverzeichnis	167