

Free- and Open-Source-Softwareentwicklung in der Automobilwirtschaft

Die Automobilindustrie durchläuft einen tiefgreifenden Wandel durch Digitalisierung, Elektrifizierung und neue Wettbewerber aus China. Während traditionelle Entwicklungsmodelle mit proprietärer Software an ihre Grenzen stoßen, bietet Free- and Open-Source-Software (FOSS) neue Chancen für kollaborative Innovation.

Diese Kurzstudie analysiert die strategische Bedeutung von FOSS in der automobilen Wertschöpfungskette und identifiziert wirtschaftliche Potenziale für Zulieferer. Auf Basis erfolgreicher Praxisbeispiele und konkreter Implementierungsstrategien werden Handlungsempfehlungen entwickelt, die besonders kleinen und mittleren Unternehmen den Einstieg in FOSS-Ökosysteme ermöglichen und ihnen helfen, vom Zulieferer zum strategischen Innovationspartner zu werden.



© sdecretet/AdobeStock

Einleitung

Die Automobilindustrie befindet sich in einem tiefgreifenden Wandel. Die zunehmende Digitalisierung und Vernetzung von Fahrzeugen, der Wandel hin zur Elektromobilität und das Aufkommen neuer Mobilitätskonzepte stellen die Branche vor große Herausforderungen. Neue Player aus China drängen mit innovativen und zunehmend wettbewerbsfähigen Modellen auf den Markt und verändern die Spielregeln (Wolter 2024).

Die Veränderungen in der Automobilindustrie sind insbesondere im Hinblick auf die Entwicklung von Fahrzeugsoftware deutlich erkennbar. Während traditionell Softwarefunktionen in dedizierte elektronische Steuergeräte (ECUs) wie Motormanagement, Airbag oder ESP fest eingebettet wurden, ermöglichen heute mikroprozessorbasierte Fahrzeugcomputer mit Linux-Betriebssystemen eine vollständige Hardware-Abstraktion. Diese technologische Entwicklung hat einen grundlegenden Wandel in der Softwareentwicklung zur Folge: Der Fokus verschiebt sich von der Entwicklung von projektspezifischen Einzellösungen hin zur produktzentrierten Entwicklung.

Parallel hat sich die Wertschöpfungsstruktur in vielen Bereichen verändert: Während bei klassischen Steuergeräten Tier-1-Zulieferer nach wie vor eine hohe Entwicklungstiefe besitzen, verteilt sich die Wertschöpfung bei softwaredefinierten Fahrzeugfunktionen zunehmend auf eine komplexere Supply Chain. Diese Komplexität erhöht den Bedarf an standardisierten Schnittstellen und gemeinsam genutzten Plattformen, was neue Ansätze für Softwareentwicklung und -integration erforderlich macht. Free- and Open-Source-Software (FOSS) gewinnt dadurch strategische Bedeutung: Sie ermöglicht es Automobilherstellern und Zulieferern, nicht-differenzierende Softwarekomponenten gemeinsam zu entwickeln und zu vereinheitlichen, statt Parallelentwicklungen in der Lieferkette zu finanzieren. Initiativen wie das SDV-Projekt der Eclipse Foundation, einer der weltweit größten Open-Source-Organisationen, schaffen bereits heute durch den "Code First"-Ansatz die Grundlagen für den ersten branchenweiten Software-Stack der Automobilindustrie (Expertenkreis Transformation der Automobilwirtschaft (ETA) 2023).

Gerade für kleinere Unternehmen ist FOSS nicht nur ein technologisches Konzept, sondern ein wirtschaftlicher Imperativ: Sie ermöglicht es, das einzigartige Domänenwissen der regionalen Zulieferer zu sichern und gleichzeitig durch kollaborative Entwicklung international wettbewerbsfähig zu bleiben. In einer Zeit, in der Software zunehmend zum differenzierenden Faktor wird, können Zulieferer durch aktive Teilnahme an FOSS-Projekten von passiven Beobachtern zu aktiven Gestaltern der automobilen Zukunft werden.

Um dieses Verständnis zu vermitteln werden in dieser Kurzstudie zunächst die Grundlagen von Free- and Open-Source-Software im Automobilkontext erläutert, bevor die wirtschaftlichen Potenziale und Chancen speziell für kleine und mittlere Unternehmen dargestellt werden. Anschließend werden typische Herausforderungen bei der FOSS-Implementierung beleuchtet und mit praktischen Lösungsansätzen verknüpft. Anhand konkreter Erfolgsbeispiele aus verwandten Branchen und spezifischer Handlungsempfehlungen für Unternehmen wird aufgezeigt, wie der Einstieg in die FOSS-Entwicklung gelingen kann. Die Studie schließt mit einem Ausblick auf künftige Entwicklungen und Chancen in der Automobilwirtschaft bezüglich FOSS.

Grundlagen zu FOSS in der Automobilindustrie

FOSS steht für Free- and Open-Source-Software. Der Begriff beschreibt Software, die unter speziellen Lizenzbedingungen veröffentlicht wird und zwei wesentliche Freiheiten bietet: Der Quellcode ist öffentlich zugänglich, und jeder darf die Software frei nutzen, untersuchen, anpassen und weitergeben.¹ "Free" bezieht sich hierbei primär auf Freiheit, nicht zwangsläufig auf den Preis.

Für Automobilzulieferer bedeutet FOSS konkret: Sie erhalten Zugang zu ausgereiften Softwarekomponenten, die sie in eigene Produkte integrieren können, ohne bei jeder Entwicklung "das Rad neu erfinden" zu müssen. Anders als bei proprietärer Software können sie den Code nach eigenen Anforderungen anpassen, ohne von einzelnen Anbietern abhängig zu sein. Die genauen

1 | Im Rahmen von individuellen Lizenzbedingungen.

Nutzungsrechte und -pflichten werden in den jeweiligen Open-Source-Lizenzen festgelegt. Diese Lizenzen bieten unterschiedliche Freiheitsgrade – von solchen mit minimalen Verpflichtungen bis hin zu Lizenzen, die sicherstellen, dass abgeleitete Werke ebenfalls unter freien Lizenzen veröffentlicht werden müssen.

Warum geht es nicht mehr mit proprietärer Software allein?

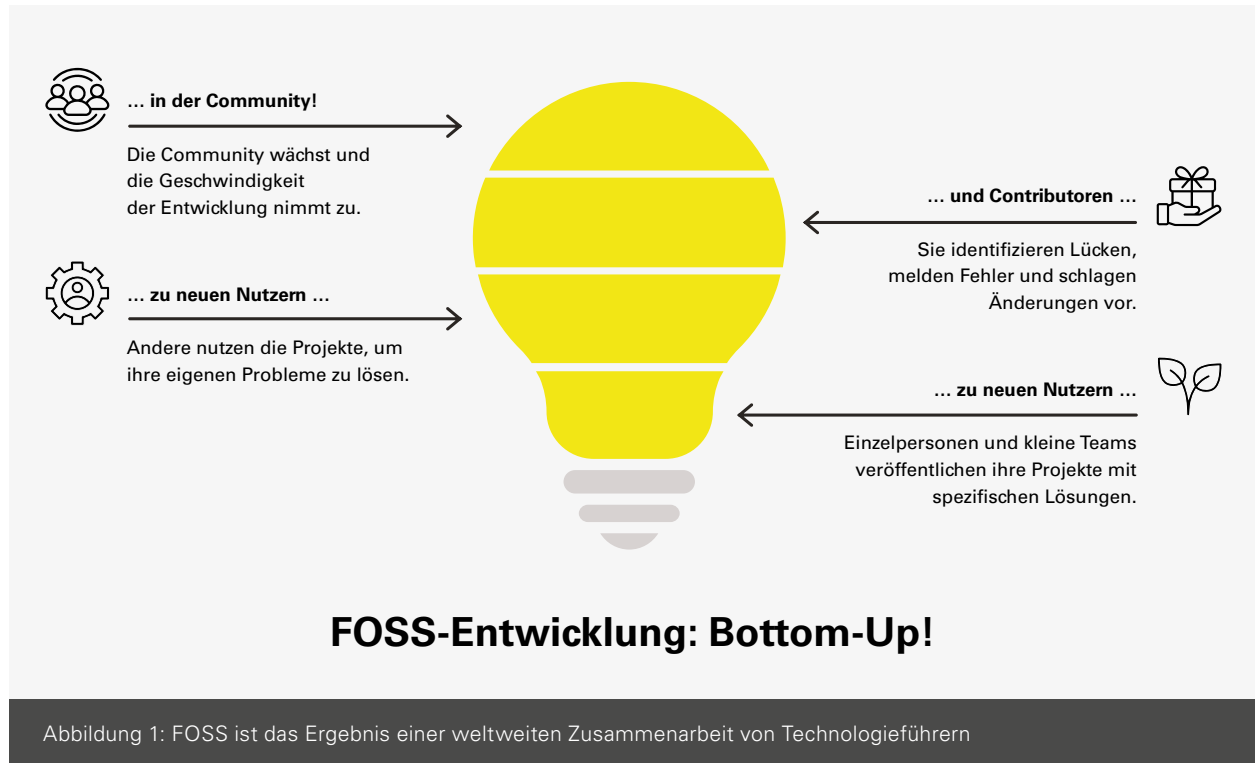
Wie unterscheidet sich die Nutzung von klassischer proprietärer Software und FOSS?	
Proprietäre Software	Free- and Open-Source-Software (FOSS)
Quellcode als geschütztes Geschäftsgeheimnis	Quellcode öffentlich zugänglich und prüfbar
Nutzung und Veränderung vertraglich eingeschränkt	Freie Nutzung, Anpassung und Weitergabe unter Einhaltung der Lizenzbedingungen
Expliziter Vertragsabschluss mit dem Anbieter erforderlich	Automatischer Lizenzvertrag durch Nutzung der Software
Typischerweise kostenpflichtige Nutzung	Meist kostenfreie Nutzung. Kosten entstehen für Anpassung, Betrieb und Wartung
Direkte Monetarisierung durch Lizenzverkauf und Softwareupdates	Monetarisierung durch Beratung, Support, Integration, Anpassung und ergänzende proprietäre Module
Haftung, Gewährleistung und Support durch Rechteinhaber	Haftung und Gewährleistung ausgeschlossen, Support durch Community oder Dienstleister
Anbieterabhängigkeit (Vendor Lock-in)	Technologische Souveränität und Flexibilität
Kundenwünsche werden nach Priorität des Anbieters umgesetzt	Eigene Anpassungen und Weiterentwicklungen jederzeit möglich

In der heutigen vernetzten Fahrzeuglandschaft stößt proprietäre Software zunehmend an ihre strukturellen Grenzen. Die Transformation zum Software-definierten Fahrzeug (SDV) verändert fundamentale Marktmechanismen, wodurch auch mittelständische Unternehmen ihre Softwarestrategie neu ausrichten müssen. Der wachsende Wertanteil von Software in Fahrzeugen steigt kontinuierlich, während sich gleichzeitig die Innovationszyklen verkürzen. Traditionelle Entwicklungsprozesse können mit dem Innovationstempo der SDVs nicht mehr Schritt halten. Ein funktionierendes Update-Management über den gesamten Produktlebenszyklus wird zum entscheidenden Wettbewerbsfaktor, den proprietäre Einzellösungen nur mit unverhältnismäßig hohem Ressourceneinsatz leisten können.

Besonders problematisch dabei ist die Abhängigkeit von einzelnen Anbietern, die die technologische Souveränität sowohl von OEMs als auch von kleineren Zulieferern gefährdet (Vendor Lock-in). Die proprietären Entwicklungsprozesse führen darüber hinaus zu fragmentierten Softwarearchitekturen: Jeder Zulieferer entwickelt seine Komponenten mit eigenen Schnittstellen, Kommunikationsprotokollen und Dateiformaten. Diese mangelnde Interoperabilität erzeugt erhebliche Mehrkosten bei Integration und Test und erschwert die Realisierung fahrzeugübergreifender Funktionen – ein Problem, das mit der steigenden Anzahl vernetzter Systeme im Fahrzeug exponentiell zunimmt. Die vermeintlichen Vorteile des geschützten geistigen Eigentums werden zunehmend von der Marktrealität überholt: Im rasanten Tempo der Branchenentwicklung verliert proprietäre Software schnell an Wert, da der Innovationszyklus sich stetig verkürzt. Währenddessen riskieren Unternehmen, die an geschlossenen Systemen festhalten, dass ihre geschützten Einzellösungen von gemeinschaftlich entwickelten Open-Source-Standards überholt werden – ein strategisches Risiko, das sich weder OEMs noch kleinere Zulieferer langfristig leisten können. Die zentrale Herausforderung für Unternehmen liegt daher nicht in der Frage, ob FOSS eingesetzt werden sollte, sondern in der strategischen Entscheidung, welche Komponenten als differenzierende Kernkompetenz proprietär bleiben sollten und wo die Teilnahme an FOSS-Ökosystemen langfristige Wettbewerbsvorteile sichert.

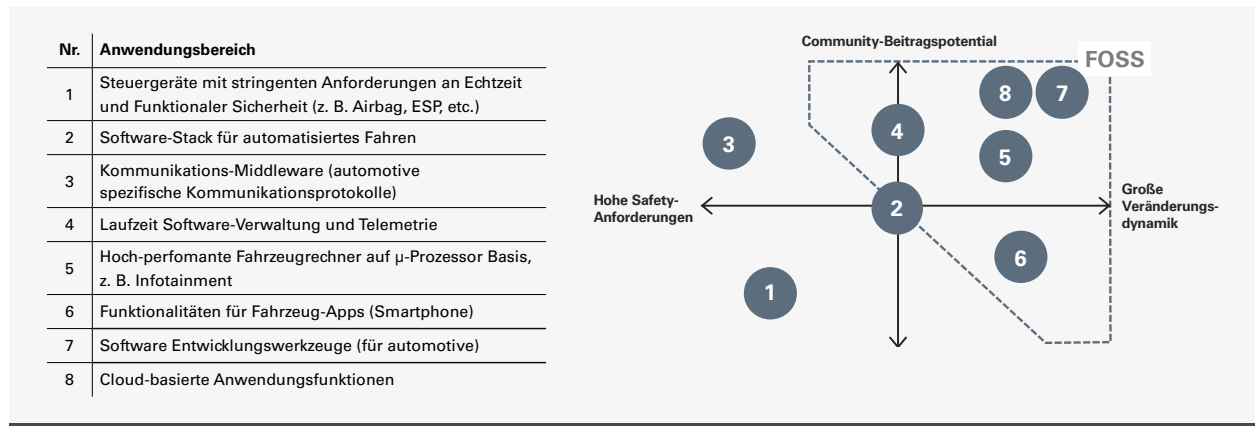
Welche Akteure sind an der FOSS-Entwicklung beteiligt?

Die FOSS-Entwicklung im Automobilbereich folgt einem grundlegend anderen Muster als traditionelle Software-Entwicklung. Statt einer Top-down-Planung durch OEMs mit anschließender Umsetzung durch Zulieferer entsteht FOSS typischerweise von unten nach oben ("bottom-up"):



Die größten automotiven FOSS-Projekte werden heute von industriegeführten Konsortien wie AUTOSAR oder der Eclipse Foundation (SDV-Projekt, mehr dazu unten) koordiniert, in denen sowohl OEMs als auch Zulieferer aller Größenordnungen zusammenarbeiten.

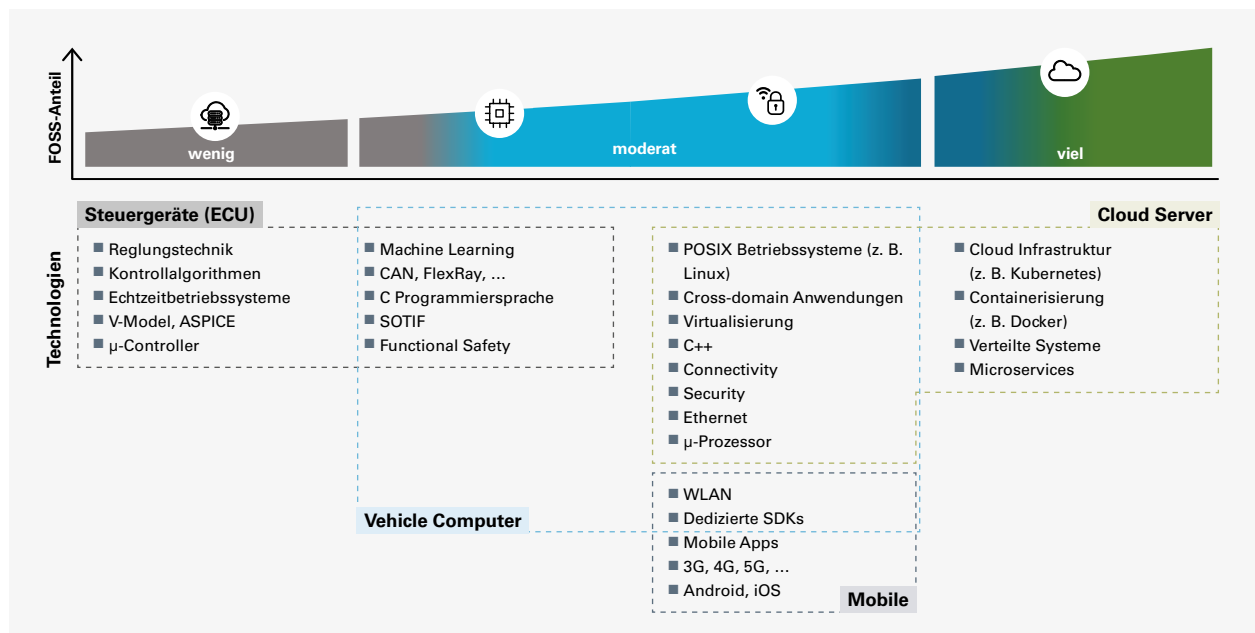
Wo wird FOSS bereits eingesetzt?



Quelle: e-mobilität BW/eigene Darstellung

Abbildung 2: FOSS eignet sich besonders in Anwendungsbereichen, die gemeinschaftlich entwickelt werden und großer Veränderungsdynamik unterliegen

FOSS eignet sich besonders für Anwendungsbereiche, die von gemeinschaftlicher Entwicklung profitieren und einer hohen Veränderungsdynamik unterliegen. In der Automobilbranche zeigt sich ein klares Muster: Je höher die Anforderungen an funktionale Sicherheit (Safety) und je tiefer die Software in der Hardware verankert ist, desto geringer ist typischerweise der FOSS-Anteil. Bei Steuergeräten mit stringenten Anforderungen an Echtzeit und funktionaler Sicherheit (wie Airbag- oder ESP-Systemen) dominiert weiterhin proprietäre Software. Diese muss hochspezialisiert auf die zugrundeliegende Mikrocontroller-Hardware optimiert sein und in kritischen Situationen zuverlässig funktionieren, um Menschenleben zu schützen. Der FOSS-Anteil steigt dagegen deutlich in Bereichen mit geringeren Safety-Anforderungen: Bei Infotainment-Systemen, Cloud-Anwendungen und Entwicklungswerkzeugen ist FOSS bereits heute Standard. Die folgende Grafik verdeutlicht, wie der FOSS-Anteil mit abnehmenden Safety-Anforderungen und zunehmender Veränderungsdynamik wächst.



Quelle: e-mobilität BW/eigene Darstellung

Abbildung 3: Wo steckt (heute schon) wie viel FOSS drin?

Wirtschaftliche Potenziale und Chancen für Unternehmen

Wie lässt sich die Zusammenarbeit mit OEMs durch FOSS verbessern?

Die traditionelle Geschäftsbeziehung zwischen Automobilherstellern und Zulieferern wandelt sich durch Free- and Open-Source-Software (FOSS) grundlegend. Statt der klassischen hierarchischen Struktur, bei der OEMs detaillierte Anforderungen vorgeben und Zulieferer diese projektspezifisch umsetzen, ermöglicht FOSS eine partnerschaftlichere Zusammenarbeit. Durch gemeinsame Entwicklungsplattformen entsteht ein kollaboratives Ökosystem, das für beide Seiten Vorteile bietet. Ein wesentlicher Aspekt dieser Transformation ist die Entwicklung einheitlicher technischer Schnittstellen durch Open-Source-Projekte. Diese ermöglichen eine bessere Interoperabilität zwischen Komponenten verschiedener Hersteller und reduzieren die Komplexität der Integration. Die Transparenz des offenen Quellcodes fördert dabei eine klare technische Kommunikation und einen effektiven Wissensaustausch zwischen allen Beteiligten.

Für Zulieferer bietet dieser Ansatz nicht nur technologische, sondern auch wirtschaftliche Vorteile: Durch standardisierte Schnittstellen reduziert sich der Integrationsaufwand für neue Komponenten erheblich, was direkte Kosteneinsparungen bedeutet. Die Beteiligung an relevanten FOSS-Projekten ermöglicht zudem frühzeitigen Zugang zu technischen Entwicklungen und Markttrends, wodurch Zulieferer ihre Produkte proaktiv anpassen können. Dies verringert das Risiko kostspieliger Nachentwicklungen und sichert die Anschlussfähigkeit an zukünftige OEM-Anforderungen. Größere Unternehmen wie Mercedes-Benz, Porsche und Continental-Automotive haben ihr Commitment zu FOSS bereits in eigenen Open-Source-Manifestos verkündet.

Der branchenübergreifende Wissensaustausch wird zudem durch regionale Communities wie The FOSS-LÄND Community gefördert, die wichtige Plattformen für Vernetzung und Kooperation bieten. Diese Entwicklung hin zu einer kollaborativen Zusammenarbeit zwischen OEMs und Zulieferern wird auch vom Expertenkreis Transformation der Automobilwirtschaft als entscheidender Erfolgsfaktor für die Zukunft der Branche hervorgehoben (Expertenkreis Transformation der Automobilwirtschaft (ETA) 2023).

Wie können Unternehmen durch FOSS Entwicklungskosten senken?

- **Vollständige Transparenz des Quellcodes:** Ermöglicht gründliche Qualitäts- und Sicherheitsprüfungen.
- **Kosteneinsparungen:** Durch die Nutzung bereits entwickelter und getesteter Komponenten können sich Unternehmen auf Verbesserungen und Erweiterungen konzentrieren, anstatt ganze Systeme neu aufzubauen.
- **Hohe Unabhängigkeit:** Keine Bindung an einen einzelnen Softwareanbieter.
- **Kürzere Entwicklungszyklen (Time to Market):** FOSS beschleunigt die Entwicklung durch die Bereitstellung vorhandener Komponenten und Werkzeuge.
- **Hoher Reifegrad der Software:** Kontinuierliche Weiterentwicklung, Förderung von Innovationen und Zusammenarbeit mit Technologieführern.
- **Community-Support:** Aktive FOSS-Gemeinschaften bieten wertvollen Support und Fachwissen.

Welche neuen Geschäftsmodelle entstehen?

Für FOSS in der Automobilbranche stechen zwei besonders relevante Geschäftsmodelle hervor: Open Core und Dual Licensing. Beim Open Core-Modell wird eine Basisversion der Software kostenlos als Open-Source angeboten, während erweiterte Funktionen oder spezielle Module kommerziell lizenziert werden (Umm-e-Laila, Najeed Ahmed Khan, und Asad Arfeen 2023). Dieses Modell ermöglicht es Zulieferern, einerseits von der Community-Entwicklung zu profitieren und andererseits Premiumfunktionen gewinnbringend zu vermarkten. So können beispielsweise grundlegende Diagnosewerkzeuge offen zur Verfügung gestellt werden, während fortgeschrittene Analysen oder spezifische Sicherheitsfunktionen kostenpflichtig bleiben.

Das Dual Licensing-Modell bietet denselben Software-Code unter zwei verschiedenen Lizenzen an: einer freien Open-Source-Lizenz und einer kostenpflichtigen kommerziellen Lizenz (Hall 2017). Dabei ist die Open-Source-Version oft mit Bedingungen verbunden, die verlangen, dass bei Weiterverwendung der Software auch der eigene Code als FOSS veröffentlicht werden muss. Dies kann für manche Kunden ein Hindernis darstellen. Die kommerzielle Lizenz hebt diese Einschränkung auf und erlaubt es Kunden, die Software in ihre eigenen Produkte zu integrieren, ohne ihr geistiges Eigentum offenlegen zu müssen. So bleibt die Kontrolle über das geistige Eigentum gewahrt, während gleichzeitig die Vorteile der Open-Source-Community für die Entwicklung genutzt werden.

Ein weiteres Geschäftsmodell, das zusätzlich zu den bereits genannten eingesetzt werden kann, ist das Anbieten von Support- und Wartungsdienstleistungen für FOSS-Produkte. Konkret können Zulieferer technischen Support, Fehlerbehebung, Kompatibilitätsprüfungen und regelmäßige Sicherheitsupdates als kostenpflichtige Leistungen anbieten. Als Entwickler der ursprünglichen Komponenten verfügen sie über einzigartiges Wissen, das für den reibungslosen Betrieb und die Integration der Software unverzichtbar ist. Diese Expertise lässt sich durch unterschiedliche Supportpakete mit garantierten Bearbeitungszeiten für Supportanfragen und definierten Leistungsumfängen gewinnbringend vermarkten – ähnlich wie es bei der Hardware-Wartung von Produktionsanlagen bereits etablierte Praxis ist. Obwohl die Software selbst kostenfrei verfügbar ist, zahlen Automobilhersteller gerne für die Gewissheit, dass ein kompetenter Partner bei Problemen schnell und zuverlässig unterstützt. Besonders für mittelständische Zulieferer stellt dieses Geschäftsmodell eine nachhaltige Möglichkeit dar, über den gesamten Produktlebenszyklus hinweg konstante Einnahmen zu generieren und gleichzeitig enge Kundenbeziehungen aufzubauen.

Herausforderungen und Lösungsansätze

Aus unternehmerischer Sicht ist es riskant, nicht zu wissen, ob bzw. welche Free- and Open-Source-Software (FOSS) in den eigenen Produkten verbaut ist. Da FOSS stets mit rechtlich bindenden Lizenzen einhergeht, müssen diese auch entsprechend befolgt und eingehalten werden. Open-Source ist kein rechtsfreier Raum. Dies muss jedem bewusst sein. Aus diesem Grund ist es zwingend notwendig, stabile, toolgestützte Abläufe zu etablieren, mit deren Hilfe Open-Source-Komponenten in den eigenen Produkten identifiziert werden können. Für alle Open-Source-Anteile müssen die Lizenzbedingungen bekannt sein und eingehalten werden. Auf technischer Ebene erfolgt dies durch Software Bills of Materials (SBOMs). SBOMs funktionieren ähnlich wie die in der Automobilindustrie bekannten Stücklisten: Sie dokumentieren präzise, welche Softwarekomponenten in einem Produkt verbaut sind, einschließlich aller Abhängigkeiten und deren Versionen. Es gibt automatisierte Werkzeuge (z. B. OSS Review Toolkit, FOSSology oder SPDX), die diese SBOMs erstellen und die Lizenzkompatibilität analysieren können (Kengo Oka 2021, 91-109).

Glücklicherweise finden Unternehmen bei diesen Herausforderungen zahlreiche etablierte Unterstützungsangebote. Initiativen wie die Linux Foundation OpenChain bieten standardisierte Prozesse und Best Practices für ein wirksames Open-Source-Compliance-Programm. Werkzeuge wie OSS Review Toolkit (ORT) und FOSSology unterstützen bei der automatisierten Erkennung und Dokumentation von Open-Source-Komponenten. Regionale Netzwerke wie The FOSS-LÄND Community in Baden-Württemberg dienen als Anlaufstellen für Unternehmen, die einen strukturierten Einstieg in die Open-Source-Entwicklung suchen. Dennoch benötigt jedes Unternehmen Mitarbeiter mit Fachkenntnissen zu Open-Source-Lizenzen, um die rechtlichen Anforderungen korrekt zu interpretieren und umzusetzen. Die Produktmanager müssen sich darüber im Klaren sein, dass unterschiedliche Open-Source-Lizenzen unterschiedliche Auswirkungen auf die Auslieferung der eigenen Produkte haben. So müssen zum Beispiel mit der Produktauslieferung die entsprechenden Open-Source-Lizenzen ausgewiesen werden.

Für Entscheidungsträger im Unternehmen ist es wesentlich, die grundlegenden FOSS-Lizenztypen und deren strategische Implikationen zu verstehen. Die Wahl zwischen permissiven Lizenzen (wie MIT, Apache) und Copyleft-Lizenzen (wie GPL) beeinflusst direkt, welche Geschäftsmodelle realisierbar sind und wie Innovationen geschützt werden können. Permissive Lizenzen erlauben die Integration in proprietäre Produkte ohne Offenlegungspflicht des eigenen Codes, was mehr Flexibilität für kommerzielle Anwendungen bietet. Copyleft-Lizenzen hingegen verlangen, dass Modifikationen unter gleichen Bedingungen weitergegeben werden, was die Bildung offener Ökosysteme fördert, aber bestimmte Geschäftsmodelle einschränken kann. Die strategische Entscheidung für oder gegen bestimmte Lizenzen sollte daher im Einklang mit der übergeordneten Unternehmensstrategie und dem angestrebten Wettbewerbsvorteil stehen.²

² | Eine detaillierte Übersicht und Entscheidungshilfe zu Open-Source-Lizenzen bieten Internetseiten wie choosealicense.com, eine von GitHub bereitgestellte Ressource, sowie opensource.guide für umfassende Leitfäden zur Open-Source-Praxis. Die Open-Source-Initiative (opensource.org) bietet zudem Definitionen und einen Katalog anerkannter Open-Source-Lizenzen.

Risiken und Herausforderungen	Lösungen
Strategische Fehler bei der Lizenzauswahl schränken Geschäftsmodelle ein und gefährden Produktstrategien.	Strategische Abstimmung der Lizenzstrategie mit Geschäftszielen. Management-Awareness durch regelmäßige Reviews sicherstellen.
Nichteinhaltung von Lizenzbedingungen führt zu rechtlichen Konsequenzen und Reputationsschäden.	Toolgestützte Überwachung durch SBOMs. Etablierung von Freigabeprozessen. Entwickler und Team-Leads entsprechend schulen.
Teilweise eingeschränkter Support, vor allem bei kleinen Projekten.	Aktive Beteiligung (Knowledge Aufbau), Projekte strategisch auswählen, Drittanbieter Support.
FOSS-Projekte können sich in eine ungewollte Richtung oder gar nicht mehr weiterentwickeln.	Aktiv mit eigenen Interessen in die Diskussion einbringen. Ggf. Projekte forken (eine eigene Kopie erstellen) und selbst nach eigenen Anforderungen weiterentwickeln.
Nichteinhaltung von Qualitätssicherungsstandards ohne vertragliche FOSS-Qualitätsgarantie.	Implementierung eigener QS-Prozesse mit systematischen Code-Reviews, Test-Suiten und Dokumentation zur Erfüllung von Zertifizierungsanforderungen.
Sicherheitslücken können zu Datendiebstahl führen (nicht FOSS spezifisch).	Softwareabhängigkeiten und Code regelmäßig aktualisieren.

Eclipse Software Defined Vehicle (SDV)

Die Eclipse Software Defined Vehicle (SDV) Working Group ist ein von der Industrie geführtes Konsortium, das die Entwicklung von SDV-Lösungen durch offene Zusammenarbeit beschleunigen will. Sie agiert als herstellernerneutrale Organisation, die gleiche Wettbewerbsbedingungen für alle Teilnehmer gewährleistet. Die Arbeitsgruppe kultiviert eine dynamische Open-Source-Gemeinschaft, in der die Mitglieder frei Wissen austauschen, Code beisteuern und gemeinsam die Komplexität der Entwicklung der nächsten Fahrzeuggeneration angehen können.

Im Kern konzentriert sich die Arbeitsgruppe auf die Schaffung einer robusten, modularen und anpassungsfähigen Open-Source-Plattform für Fahrzeugsoftware. Diese Plattform umfasst eine Vielzahl von Projekten, die sich mit verschiedenen Aspekten der SDV-Entwicklung befassen, von Software-Stacks im Fahrzeug und Cloud-basierten Mobilitätslösungen bis hin zu umfassenden Entwicklungs-Toolchains und optimierten Arbeitsabläufen (Eclipse Foundation AISBL 2025).

In einem bedeutenden Schritt in Richtung einer industrieweiten Zusammenarbeit haben sich vier wichtige Branchenkonsortien zusammengeschlossen, um die SDV Alliance zu gründen: AUTOSAR (ein etabliertes Konsortium für Automotive-Softwarearchitektur), COVESA (Connected Vehicle Systems Alliance), Eclipse SDV (die bereits beschriebene Software Defined Vehicle Working Group) und SOAFEE (eine Initiative für skalierbare Fahrzeugsoftwarearchitekturen). Diese Allianz zielt darauf ab, die Entwicklung von Software-definierten Fahrzeugen zu vereinfachen und zu beschleunigen, indem sie die Zuständigkeiten der einzelnen Organisationen klar definiert und die Zusammenarbeit zwischen diesen bisher getrennten Initiativen fördert (Lampinen 2024).

Praxisbeispiele

In diesem Kapitel wird anhand konkreter Erfolgsbeispiele aus verschiedenen Branchen gezeigt, wie Unternehmen FOSS gewinnbringend einsetzen. Diese Beispiele dienen als Orientierung und Motivation für Zulieferer und explizit auch für kleinere Unternehmen.

Die folgende Tabelle präsentiert fünf herausragende Beispiele für erfolgreiche FOSS in unterschiedlichen Technologiebereichen. Diese Projekte demonstrieren, wie FOSS nicht nur von großen Unternehmen entwickelt und genutzt wird, sondern auch erhebliches Potenzial für kleine und mittlere Unternehmen (KMUs) bietet. Jedes der vorgestellten Projekte hat in seiner Branche signifikanten Einfluss erzielt und zeigt, wie FOSS Innovation fördert, Markteintrittsbarrieren senkt und nachhaltige Ökosysteme schafft. DeepSeek Inc. hatte zum Zeitpunkt der Veröffentlichung ihres Produkts sogar weniger als 200 Mitarbeiter und würde damit als KMU gelten.³ Reacts Entwicklung begann mit nur einem Entwickler bei Facebook und konnte inzwischen zu einem Ökosystem mit Millionen Entwicklern gedeihen. Auch in der Automobilbranche kristallisieren sich inzwischen größere Projekte wie Automotive Grade Linux (AGL) oder Baidu Apollo heraus.

	Beschreibung	Verwendung	Nutzen
DeepSeek R1	Open-Source-Sprachmodell mit sehr gutem Preis-Leistungs-Verhältnis.	Beeindruckende Leistung in Benchmarks und etabliert als konkurrenzfähige Alternative zu proprietären Modellen wie ChatGPT.	Ermöglicht günstigen Zugang zu fortschrittlicher KI für Anwendungen wie Chatbots, Content-Erstellung und Automatisierung.
Facebook (Meta) React	Open-Source-JavaScript-Bibliothek zur Entwicklung moderner, interaktiver Benutzeroberflächen für Web- und mobile Anwendungen.	Weltweiter Einsatz durch tausende Unternehmen wie Netflix und Airbnb mit einem umfangreichen Ökosystem an Erweiterungen.	Ermöglicht die Entwicklung hochwertiger Webanwendungen mit konsistenter Darstellung auf verschiedenen Plattformen.
Automotive Grade Linux	Einheitliches Open-Source-Betriebssystem für Fahrzeuginfotainment, Telematik und vernetzte Funktionen.	Nutzung durch führende Automobilhersteller wie Toyota und Mercedes-Benz mit nachweislicher Senkung der Entwicklungskosten.	Bietet eine kostenfreie Basis für Automotive-Lösungen und verbessert ihre Wettbewerbsfähigkeit im OEM-Umfeld.
Baidu Apollo	Open-Source-Plattform für autonomes Fahren mit Integration von KI, Sensorik und Echtzeitdatenverarbeitung.	Über 200 Partner wie BMW und NVIDIA nutzen Apollo für kommerzielle Lösungen wie Robotaxis und autonome Minibusse.	Bietet Zugang zu hochmodernen Algorithmen für autonome Fahrsysteme und reduziert Eintrittsbarrieren im Markt.
Robot Operating System (ROS)	Framework für die Entwicklung von Robotik-Software mit Bibliotheken und Entwicklungswerkzeugen für diverse Robotik-Anwendungen.	De-facto Standard in der Robotikforschung mit zunehmender industrieller Nutzung durch BMW und Tier-1-Zulieferer.	Ermöglicht kostengünstigen Einstieg in die Robotik-Entwicklung mit umfangreicher Community-Unterstützung.

³ | Umsatzzahlen für 2024 sind nicht öffentlich einsehbar, da DeepSeek Inc. ein chinesisches Privatunternehmen ist. Es ist jedoch anzumerken, dass High-Flyer, die private Investment Management Firma, die DeepSeek besitzt, mit 160 Mitarbeitern (2021) sieben Milliarden US\$ verwaltetes Vermögen angibt (Oktober 2024).

Handlungsempfehlungen für Unternehmen

Der Einstieg in Free- and Open-Source-Software (FOSS) erfordert einen systematischen Ansatz, der über die bloße Nutzung "kostenloser Software" hinausgeht. Im ersten Schritt sollten Unternehmen ihre Softwarekomponenten analysieren und kategorisieren: Welche Bereiche sind wettbewerbsdifferenzierend und müssen intern entwickelt werden? Wo können standardisierte FOSS-Lösungen zum Einsatz kommen? Parallel dazu ist es essentiell, rechtliche Rahmenbedingungen zu klären - insbesondere Lizenz- und Haftungsfragen. Da kleine und mittlere Unternehmen selten über spezialisierte Rechtsabteilungen verfügen, empfiehlt sich hier die Zusammenarbeit mit externen Experten oder Branchenverbänden.

Die Mitarbeiterqualifizierung bildet einen weiteren Kernaspekt: Teams müssen sowohl mit den technischen als auch organisatorischen Besonderheiten von Open-Source-Entwicklung vertraut gemacht werden. Dies betrifft nicht nur Entwickler, sondern auch Projektmanager und Führungskräfte. Besonders wichtig ist das Verständnis, dass die Pflege und Integration von FOSS den gleichen professionellen Aufwand erfordert wie proprietäre Software - mit dem Unterschied, dass sich dieser Aufwand durch die breite Nutzung in der Community amortisiert.

Für die praktische Umsetzung empfiehlt sich ein schrittweiser Einstieg: Zunächst durch aktive Teilnahme an bestehenden Open-Source-Communities, um Erfahrungen zu sammeln und von etablierten Best Practices zu lernen. Automotive-spezifische Initiativen wie die Eclipse Foundation bieten hier einen guten Startpunkt. Mit wachsender Erfahrung können Unternehmen dann eigene Beiträge leisten und die Vorteile der kollaborativen Entwicklung ausschöpfen. Dies schließt auch die Möglichkeit ein, durch gezielte Zusammenarbeit in den Communities die Entwicklung in eine für das eigene Geschäftsmodell nützliche Richtung zu lenken.

	Auswahl des richtigen Geschäftsmodells: Die Wahl eines Modells, das zu der Software und dem Zielmarkt passt, ist entscheidend. Dies könnte bedeuten, Premium-Support anzubieten, ein Software-as-a-Service-Angebot zu schaffen oder einen dualen Lizenzierungsansatz mit einer kommerziellen Version neben der Open-Source-Version zu wählen.
	Engagement der Community: Der Aufbau einer starken Community rund um die Software fördert die Akzeptanz und regt zu Beiträgen an. Transparenz, klare Beitragsrichtlinien und aktives Engagement sind dabei unerlässlich.
	Verstehen der Lizenzierung: Gründliche Kenntnisse der Open-Source-Lizenzen sind entscheidend, um rechtliche Probleme zu vermeiden und die Einhaltung der Vorschriften zu gewährleisten.
	Bereitstellung von Mehrwertdiensten: Das Angebot von Dienstleistungen wie Beratung, Schulung oder Integration kann Einnahmequellen schaffen und das Unternehmen von der Konkurrenz abheben.
	Berücksichtigung von Sicherheitsbedenken: Besonders in sicherheitskritischen Bereichen ist es für den Aufbau von Vertrauen unerlässlich, Engagement für die Sicherheit zu demonstrieren und Schwachstellen zu beseitigen.
	Aufbau einer starken Marke: Die Schaffung einer erkennbaren Marke und die Etablierung einer Vordenkerrolle innerhalb der Open-Source-Gemeinschaft können Nutzer, Kunden und Talente anziehen.
	Konzentration auf die Nachhaltigkeit: Ein Gleichgewicht zwischen den Bedürfnissen der Gemeinschaft und der finanziellen Nachhaltigkeit ist entscheidend für den langfristigen Erfolg.

Abbildung 4: Erfolgskriterien für FOSS in Unternehmen

Quelle: e-mobil BW/ eigene Darstellung

Fazit

Diese Studie zeigt, dass Free- and Open-Source-Software in der Automobilbranche nicht mehr nur eine technische Alternative darstellt, sondern zu einem entscheidenden strategischen Faktor geworden ist. In einer Zeit des tiefgreifenden Wandels durch Digitalisierung, Elektrifizierung und neue Wettbewerber bietet FOSS den Zulieferern eine Chance, ihre Position in der Lieferkette neu zu definieren und zu stärken. Der kollaborative Entwicklungsansatz ermöglicht es beteiligten Unternehmen, gemeinsam an nicht-differenzierenden Komponenten zu arbeiten und gleichzeitig ihre Kernkompetenzen als Wettbewerbsvorteil zu nutzen.

Die Zukunftsperspektiven für FOSS in der Automobilbranche gestalten sich vielversprechend. Der vom Expertenkreis Transformation der Automobilwirtschaft beschriebene "Code First"-Ansatz wird zu einer agilen Industriestandardisierung führen, bei der Softwarekomponenten und -architekturen nicht mehr durch langwierige Abstimmungsprozesse, sondern durch praktische Implementierung und Erprobung entwickelt werden. Diese Form der Standardisierung ermöglicht es, schneller auf Marktanforderungen zu reagieren und innovative Lösungen zu etablieren.

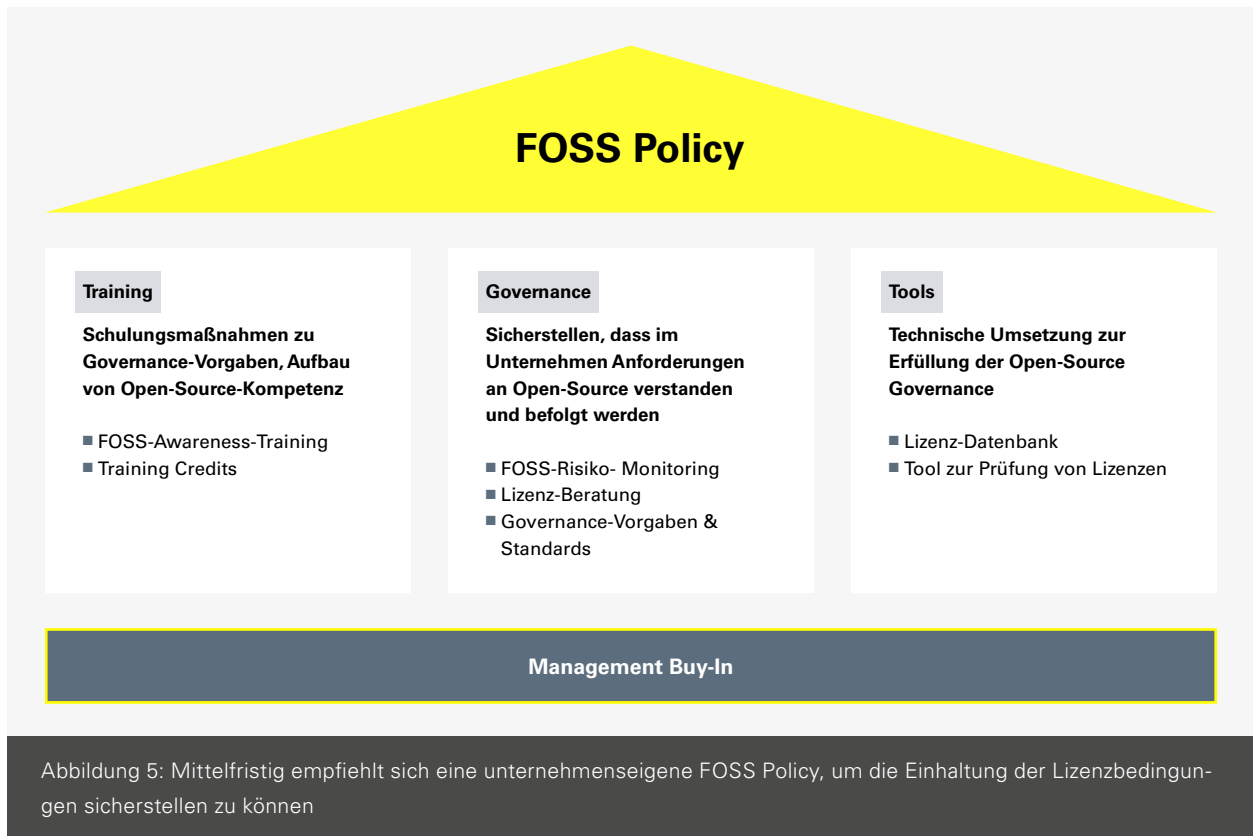
Rund um softwarezentrierte Fahrzeugplattformen werden sich neue Ökosysteme entwickeln, die auf FOSS-Technologien basieren. Diese Ökosysteme bieten insbesondere für kleine und mittlere Unternehmen neue Geschäftsfelder – sei es als Entwickler spezialisierter Module, als Integratoren oder als Anbieter von Support- und Wartungsdienstleistungen. Die traditionelle hierarchische Beziehung zwischen OEMs und Zulieferern wird sich durch FOSS zu einer partnerschaftlicheren Zusammenarbeit wandeln.

In den kommenden Jahren wird der Anteil von FOSS in Fahrzeugsoftware weiter zunehmen, wobei besonders die Bereiche Infotainment, Konnektivität und Backend-Systeme von dieser Entwicklung profitieren werden. Automobilspezifische FOSS-Communities wie die Eclipse Software Defined Vehicle Working Group werden an Bedeutung und Einfluss gewinnen und die Entwicklung branchenweiter Standards maßgeblich mitgestalten.

Um von den beschriebenen Entwicklungen zu profitieren, sollten Unternehmen einen strukturierten Ansatz für den Umgang mit FOSS etablieren. Dieser basiert auf drei grundlegenden Säulen:

- Erstens benötigen Unternehmen ein umfassendes Training ihrer Organisation, damit ein tiefgreifendes Verständnis für Open-Source entwickelt wird – einschließlich der damit verbundenen Chancen und Risiken. Dies betrifft nicht nur die technischen Teams, sondern alle Bereiche des Unternehmens, vom Management bis zur Produktentwicklung.
- Zweitens ist eine effektive Governance unerlässlich, um sicherzustellen, dass die Anforderungen, die Open-Source an das Unternehmen stellt, verstanden und konsequent befolgt werden. Dies umfasst sowohl rechtliche Aspekte wie Lizenzmanagement als auch organisatorische Fragen der Zusammenarbeit mit Communities.
- Drittens kann eine erfolgreiche FOSS-Strategie nur mit einem strukturierten, werkzeuggestützten Ansatz umgesetzt werden, der die Identifikation, Verwaltung und Compliance von Open-Source-Komponenten sicherstellt.

Das Fundament für diese drei Säulen bildet ein solides Management Buy-In, das mit einem fundierten Verständnis der Materie einhergeht. Nur wenn die Führungsebene die strategische Bedeutung von FOSS erkennt und aktiv unterstützt, kann eine unternehmensweite FOSS-Policy erfolgreich implementiert werden.



Durch die proaktive Auseinandersetzung mit FOSS können Unternehmen nicht nur ihre Wettbewerbsfähigkeit in der transformierten Automobilwirtschaft sichern, sondern auch aktiv an der Gestaltung der automobilen Zukunft mitwirken. Die Transformation bietet gerade für Unternehmen mit spezifischem Domänenwissen die Chance, sich vom reinen Zulieferer zum strategischen Partner und Innovationstreiber zu entwickeln. Durch den gezielten Einsatz von FOSS können Zulieferer in der Automobilindustrie nicht nur Lizenz- und Entwicklungskosten reduzieren, sondern auch ihre Marktposition stärken. Die wachsende Bedeutung von Software im Automobil führt dazu, dass proprietäre Alleingänge zunehmend riskant werden. FOSS-basierte Komponenten bieten hingegen einen praktischen Weg, um Interoperabilität zu gewährleisten und gleichzeitig die Flexibilität zu behalten, auf sich ändernde Anforderungen der OEMs kosteneffizient zu reagieren. Für viele Zulieferer stellt dies nicht mehr nur eine strategische Option dar, sondern entwickelt sich zur grundlegenden Voraussetzung, um wettbewerbsfähige Produkte anbieten zu können.

Referenzen

Eclipse Foundation AISBL. n.d. "Eclipse Software Defined Vehicle." Eclipse Software Defined Vehicle. Accessed 3 6, 2025. <https://projects.eclipse.org/working-group/eclipse-software-defined-vehicle>.

Expertenkreis Transformation der Automobilwirtschaft (ETA). 2023. Vom Hardware-Produkt zum Software-Defined Vehicle: Durch Open-Source-Softwareentwicklung den Automobilstandort Europa stärken, Policy Paper. N.p.: Bundesministerium für Wirtschaft und Klimaschutz. <https://expertenkreis-automobilwirtschaft.de/>.

Hall, Andrew J. 2017. "Open-Source Licensing and Business Models: Making Money by Giving it Away." Santa Clara High Technology Law Journal 33, no. 3 (2): 427–437.

Kengo Oka, Dennis. 2021. Building Secure Cars: Assuring the Automotive Software Development Lifecycle. N.p.: Wiley.

Lampinen, Megan. 2024. "Open-source software collaboration "essential" for SDVs." Automotive World. <https://www.automotiveworld.com/articles/open-source-software-collaboration-essential-for-sdvs/>.

Umm-e-Laila, F., S. Najeed Ahmed Khan, and T. Asad Arfeen. 2023. "Framework for Identification of Critical Factors for Open Source Software Adoption Decision in Mission-Critical IT Infrastructure Services." IETE Journal of Research 69 (2): 635-648. 10.1080/03772063.2021.1994036.

Wolter, Hans J. 2024. "Wettbewerbsfähigkeit der Auto-Industrie: So kommt die Branche wieder in Fahrt." Aktiv online. <https://www.aktiv-online.de/news/wettbewerbsfaehigkeit-der-auto-industrie-so-kommt-die-branche-wieder-in-fahrt-19743>.

Autor: Tim Bittner

Danksagung: Mein Dank gilt Prof. Dr.-Ing. Ina Schaefer für die wertvollen fachlichen Diskussionen und das sorgfältige Lektorat dieser Studie.

Automotive Software Collaboration BW – The FOSS-LÄND Community

Durch den Einsatz von Free- and Open-Source-Software (FOSS) profitieren Unternehmen von verkürzten Entwicklungszeiten, reduzierten Kosten und höherer Softwarequalität. Die Automotive Software Collaboration BW – The FOSS LÄND Community unterstützt Unternehmen in Baden-Württemberg die Nutzung und Entwicklung von FOSS zu etablieren, Open-Source-Kollaborationen zu erschließen, Prozesse zu optimieren und rechtssichere Standards zu schaffen.

Weitere Informationen: [Automotive Software Collaboration BW](#).

Die Initiative entstand im Rahmen der Sprint-Mission "Open-Source-Softwareentwicklung in der Automobilwirtschaft – Standortvorteile durch Kollaboration sichern" des Strategiedialogs Automobilwirtschaft Baden-Württemberg (SDA BW). Besonderer Dank gilt Herrn Detlef Zerfowski sowie der Kerngruppe der Sprint-Mission – Frau Anna Ewerlin, Herrn Massimo Frazzetta, Herrn Eduard Itrich, Herrn Daniel Krippner, Herrn Marcel Kurzmann und Herrn Christian Wege – für Ihre Unterstützung bei der Bereitstellung von Inhalten für die vorliegende Kurzstudie.

Beratungsgutschein "Transformation Automobilwirtschaft" mit Themenbezug FOSS

Der Beratungsgutschein "Transformation Automobilwirtschaft" unterstützt kleine und mittlere Unternehmen (KMU) in Baden-Württemberg bei der Einführung und Nutzung von Free- and Open-Source-Software (FOSS). Das Ministerium für Wirtschaft, Arbeit und Tourismus sowie das Ministerium für Verkehr Baden-Württemberg fördern bis zu fünf Beratungstage mit einem Fördersatz bis zu 80 Prozent. Antragsberechtigt sind KMU mit bis zu 250 Beschäftigten und Sitz oder Niederlassung in Baden-Württemberg. Der maximale Tageshöchstsatz pro Berater beträgt 1.250 Euro, sodass Unternehmen bis zu 5.000 Euro Förderung erhalten können. Anträge sind bis zum 31. Oktober 2025 möglich. Weitere Informationen und Antragsunterlagen: [Beratungsgutschein FOSS](#).

Herausgeber



Gefördert von



Layout/Satz/Illustration

markentrieb – Die Kraft für Marketing und Vertrieb

Stand

März 2025