AQI | Automotive Quality Institute

Softwareupdates Over-The-Air II

Eine Prozess- und Risikoanalyse mittels Simulation

Projektnummer	9810009
Autoren	Jonas Römer (AQI) Dr. Jürgen Großmann (Fraunhofer FOKUS)
Kontakt	jonas.roemer@aqigmbh.de
Datum	13.12.2018

Diese Veröffentlichung basiert auf dem Expertenwissen aus dem AQI und wissenschaftlicher Partner. Sie stellt einen konsolidierten Standpunkt zum entsprechenden Themenkomplex dar.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 MOTIVATION UND ZIELSETZUNG

Ziel dieses Nachfolgeprojektes ist es, die Ergebnisse aus dem Projekt „Softwareupdates Over-the-Air 1“ kritisch zu hinterfragen und um aktuelle Entwicklungen zu erweitern. Nachdem das erste Projekt vor allem aus der Recherche, Verknüpfung und Strukturierung von Teilaspekten bestand, um einen idealisierten, ganzheitlichen OTA-Updateprozess zu entwerfen. Basiert das Nachfolgeprojekt methodisch vor allem auf der Programmierung eines eigenen Simulationstools. Zur Entwicklung dieses Tools musste jede einzelne Aktivität im Detail analysiert und hinterfragt werden, um ein geeignetes Modell zu entwickeln. Im Abschluss wurde der aktuelle Stand von internationalen Standardisierungsverfahren und Gremienarbeiten betrachtet.

Ergebnis ist eine vollständige Aktualisierung der Prozessbeschreibung aus dem ersten Bericht. In den drei Phasen Generierung, Verteilung und Installation werden je 12, 11 bzw. 11 Aktivitäten beschrieben. Von diesen 34 Aktivitäten sind 6 neu, wobei 10 Aktivitäten aus dem Vorgängerprojekt entfallen sind, weil diese durch andere Aktivitäten ersetzt wurden, in anderen aufgegangen sind oder dem gewählten Abstraktionslevel nicht entsprachen. Jede dieser Aktivitäten ist beschrieben und um verschiedene Attribute ergänzt, um eine strukturierte Wissenssammlung zur OTA-Prozesskette bereitzustellen.

Mittels der Simulation wurden anschließend verschiedene Use Cases miteinander verglichen, eine kritische Pfad-Analyse durchgeführt und einer dieser Use Cases iterativ optimiert, um den Einfluss verschiedener Prozessparameter auf die Durchlaufzeit des Prozesses zu untersuchen. Bevor die Prozessbeschreibung dargestellt wird, folgt zunächst die Diskussion der im Vorgängerprojekt vorgeschlagenen Klassifizierungsansätze.

Ziel der Projektreihe ist es für die Thematik Softwareupdates Over-the-Air Transparenz durch Strukturierung zu schaffen und die Prozessfähigkeit herzustellen. Denn OTA-Updates sind nicht rein-technische, sondern komplexe Unternehmensprozesse.

2 EVALUIERUNG UND ÜBERARBEITUNG DES KLASSIFIZIERUNGSSCHEMAS

Eine systematische Klassifizierung von Softwareupdates entlang des Funktionsbereichs im Fahrzeug, dem Zweck des Updates oder den regulatorischen Vorgaben ermöglicht die Steigerung der Qualität, Robustheit und Effizienz des OTA-Updateprozesses. In Abhängigkeit der jeweiligen Updateklasse können die Anforderungen an die Prozessbausteine angepasst bzw. variiert werden. Ergänzend können (Mindest-)Standards und Unteraufgaben und -prozesse für die verschiedenen Klassen hinterlegt werden. Für das Unternehmen entstehen so Freiheitsgrade, durch die der Aufwand und die Prozesslaufzeit an die spezifischen Bedürfnisse der Klassen angepasst werden können. Darüber hinaus kann durch eine Klassifizierung auch eine Priorisierung bei der Verteilung von Updates erfolgen oder Anbieter von OTA-relevanten Dienstleistern qualifiziert werden.

2.1 Ursprüngliches Klassifizierungsschema

Im ersten OTA-Projekt wurden drei Dimensionen für eine möglichen Klassifizierung von OTA-Updates vorgeschlagen und begründet inwiefern sie ein OTA-Update beeinflussen. Die in Abbildung 1 dargestellten Dimensionen stellen das ursprüngliche Klassifizierungsschema dar.

Abbildung 1: Klassifizierungsschema

Updates können zunächst nach dem Funktionsbereich unterschieden werden, die sie adressieren. Dies spiegelt einerseits die organisatorischen Strukturen in der Automobilindustrie wieder und berücksichtigt andererseits die mechanischen und softwareseitigen Unterschiede zwischen Infotainment-, Karosserie-/Komfort- oder Fahrfunktionen. Die technische Abgrenzung zwischen den Updates spiegelt außerdem auch die mögliche Gefährdung von Insassen und/oder Umwelt wider, die als Konsequenz eines Fehlers der Funktion entstehen könnte.

Eine weitere Klassifizierung kann nach dem Zweck des Updates vorgenommen werden. Analog zu den Updates in der IT-Branche wird hier zwischen präventiven, korrektiven und additiven Softwareupdates differenziert. Neben der variierenden Information von Kunden und regulatorischen Behörden kann aus einer solchen Klassifizierung auch der Bedarf der Dringlichkeit eines Updates abgeleitet werden.

Die letzte Dimension berücksichtigt ausschließlich die Schnittstelle zur regulatorischen Behörde und damit die Typgenehmigungs- und/oder Zulassungsrelevanz. Bei einer solchen Abhängigkeit sind zusätzliche Anforderungen an den Update-Prozess zu richten.

Bei Anwendung jeder dieser drei Klassifizierungen können bereits Prozessvariationen entstehen, die einen Mehrwert bei der Durchführung eines Updates bieten. Es kann innerhalb einer Klassifizierung eine Dringlichkeit des Updates abgeleitet werden, der Informationsbedarf gegenüber dem Kunden oder auch eine Abstufung des Umfangs der Validierung der Funktionalität nach der Installation beschrieben werden.

Letzteres trifft jedoch auch auf zwei Klassifizierungen zu: der nach dem Update-Ziel und der nach dem Funktionsbereich. So reicht bei Softwarepflege oder Infotainment-Updates eine Überprüfung der vollständigen Installationsroutine aus, während bei Fehlerbehebungen und Fahrfunktionen eine Überprüfung der Funktion selbst im Anschluss an die Installation stattfinden sollte.

Bereits eine Einstufung von Softwareupdates nach einer dieser drei Klassifizierungen schafft einen Mehrwert für die Bearbeitung des OTA-Update-Prozesses.

2.2 Problemstellung

Bei der Einteilung von Updates nach nur einer dieser Klassifizierung kann jedoch nicht allen Anforderungen an die einzelnen Prozessaktivitäten Rechnung getragen werden. Während eine Klassifizierung für eine Anforderung eine Abstufung anbietet, muss diese Anforderung bei einer anderen Klassifizierung nicht berücksichtigt werden. Die Klassifizierung entlang einer Dimension ist daher nicht ausreichend, um alle notwendigen Prozessanforderungen zu berücksichtigen. Durch die Unabhängigkeit der Dimensionen der Klassifizierung gehen Potentiale der Prozessoptimierung verloren.

Zum Beispiel kann ein Update eine Fehlerbehebung im Infotainment oder eine Fehlerbehebung einer Fahrfunktion sein. Letztere Fehlerbehebung kann relevant für die Produktsicherheit sein, weshalb sie (a) schneller durchgeführt werden sollte, d. h. zeitkritischer ist, und (b) ggf. meldepflichtig ist. Würde man diese Anforderungen nun auch auf eine Fehlerbehebung im Infotainment (z B. nicht funktionierende GUI-Elemente) aufgrund einer Klassifizierung nach dem Update-Zweck anwenden, würden hier Potentiale bei der Durchlaufzeit und den Kosten nicht genutzt werden, weil stets die höchsten Anforderungen durch eine mögliche Safety-Relevanz genutzt werden müssten. Erst durch eine Verknüpfung beider Klassifizierungen miteinander ließen sich die Potentiale nutzen. Darüber hinaus kann es Fehlerbehebungen im Infotainment geben (z. B. fehlende Informationsanzeige über den Fahrzeugzustand), die dennoch eine zeitkritische Safety-Relevanz besitzen. Dies wäre weder mit den unabhängigen noch mit verknüpften Dimensionen eindeutig abbildbar. Vielmehr müsste man zumindest noch die dritte Dimension ergänzen, da das Update ggf. meldepflichtig wäre und so die Behördensicht relevant wird.

Bei einer vollständigen Betrachtung und logischen Verknüpfung aller möglichen Klassifizierungen könnten zwölf oder mehr Prozessvariationen folgen. Eine Klassifizierung wäre insofern nicht mehr sinnvoll, da dies zu viele Variationen sind. Daher wird in diesem Abschnitt untersucht, ob es zwischen den vielen möglichen Klassifizierungskombinationen Gemeinsamkeiten gibt, so dass die Anzahl auf wenige Prozessvariationen runtergebrochen werden kann. Ziel ist eine harmonisierte Klassifizierung mit wenigen Klassen.

2.3 Evaluierungsmethode

Zur Identifikation überflüssiger Klassen und zur Harmonisierung des gesamten Klassifizierungsschemas wurde das Klassifizierungsschemas durch Anwendung auf die gesamte Prozesskette, wie es im ersten OTA-Projekt entwickelt wurde, analysiert und validiert. Systematisch wurden die einzelnen Aktivitäten in Abhängigkeit der Klasse diskutiert. Dabei wurde analysiert, ob sich für die einzelnen Aktivitäten durch die Klassen neue Anforderungen ergeben. Gäbe es für einen definierten Satz von Aktivitäten bei mehreren Klassen die gleichen Anforderungen, wäre dies ein Indiz, diese Klassen zusammenzufassen.

2.4 Ergebnis der Evaluierung

Zunächst lässt sich festhalten, dass vor allem in der ersten Prozessphase „Generierung“ die Prozessaktivitäten aus Sicht des gesamten Klassifizierungsschemas keine Unterschiede aufweisen. Die Anforderungen, die sich in Abhängigkeit einer Klasse ändern würden, richten sich an etablierte Prozesse, wie die Entwicklung von Updates, aber nicht an den Prozess der Verteilung eines Updates Over-the-Air selbst.

Weiter lässt sich erkennen, dass die Schnittstelle zur regulatorischen Behörde (z. B. bei Rückrufen) nur ergänzende Aktivitäten bedeutet. Andere Aktivitäten der OTA-Prozesskette werden durch diese Schnittstelle nicht variiert.

Für die weiteren Aktivitäten gibt es keine Kombinationen, durch die sich die Klassifizierungen harmonisieren und auf eine geringe Anzahl reduzieren lassen. Die Kombinationen sind mehrfach und nicht eindeutig. Eine Zuordnung zu einer neuen Klassifizierung würde letztlich nur einen Kompromiss darstellen. Dadurch könnte die gewünschte Effizienzsteigerung nicht erreicht werden und der notwendigen Qualitätsabsicherung wird nicht ausreichend Rechnung getragen, da z. B. benötigte Kennzahlen nicht präzise genug definiert werden könnten. Die gewünschte Reduzierung der Komplexität (bei Verständnis, Durchführung, etc.) ist nicht erreichbar.

2.5 Alternative Klassifizierung entlang von Attributen

Viel mehr macht es Sinn dem Update Eigenschaften zuzuweisen, anhand derer weitere Anforderungen an Aktivitäten definiert werden. Neben „einfachen“ Updates können folgende Attribute Softwareupdates zugewiesen werden:

- **Zeitkritisch**
- **Sicherheitskritisch**
  - **Safety-relevant**
  - **Security-relevant**
- **Behörden-relevant**
- **Obligatorisch oder Optional**

Dabei können einem Update kein, ein oder mehrere Attribute zugewiesen werden. Nachfolgend werden deren Auswirkungen und Beispiele genauer beschrieben.

Das Attribut Behörden-relevant ermöglicht zu differenzieren, wann die Regulatorische Behörde eingebunden werden muss. Dies wäre der Fall bei einem mangelhaften Produkt, von dem eine

Gefährdung ausgeht. Dies betrifft wahrscheinlich viele Updates, auf die auch das Attribut Safety-relevant zutrifft. Dieses Attribut kann aber auch Anwendung bei additiven Updates finden (also Funktionserweiterungen), die die Fahrfunktion betreffen und bei denen Fehler in der Installationsroutine möglicherweise sicherheitskritische Auswirkungen hätten, die vorab ausreichend abgesichert werden müssen. Die Attribute beziehen sich immer auch auf das Update-Ziel – also z. B. eine Safety-relevante ECU. Sonst wäre grundsätzlich auch jedes Update Security-relevant, da der Übertragungsweg ja geschützt werden muss. In diesem Bericht ist ein solches Updates aber als Aktualisierung der Security-Mechanismen im Fahrzeug zu verstehen (siehe unten).

Durch diese Art einer ‚Klassifizierung' von Updates durch Attribute, entstehen notwendige Freiräume bei der Bewertung von Updates, so dass z. B. auch zwei verschiedene Updates von Karosseriefunktionen unterschiedlich bewertet werden können, wenn es das eine Mal eine Fehlerbehebung und das andere Mal eine Funktionsverbesserung ist. Die im Folgenden beschriebenen Update-Aktivitäten können so mit Bedingungen ergänzt werden, die den Attributen zugeordnet werden und bei entsprechenden Updates auszuführen sind. Hierdurch kann der Prozess für die verschiedenen Updatetypen mit ihren spezifischen Randbedingungen optimiert werden.

So kann beispielsweise– je nach Firmenpolitik – das zeitkritische Update schon bei Freischaltung auf das Kundenfahrzeug heruntergeladen werden, um direkt installiert zu werden. Während bei nicht zeitkritischen Updates das Herunterladen erst nach Zustimmung des Kunden erfolgt, wenn sich das Fahrzeug z. B. in einem W-LAN befindet. Dies ist letztlich ein Kostenfaktor, der aber einen Mehrwert für den Kunden schafft, da die Zeit für das Update nur noch aus der Installationszeit und nicht mehr aus Download-Zeit und Installationszeit besteht. Die Akzeptanz zur zeitnahen Durchführung eines solchen Updates kann dadurch steigen.

	zeitkritisch	Security-relevant	Safety-relevant	Behörden-relevant	Obligatorisch
Update einer Infotainment App oder Softwarepflege					
Härten des Security-Systems		X			
Schließen einer Sicherheitslücke, z. B. mit akuter Bedrohung von personenbezogener Daten, die im Fahrzeug gespeichert sind	X	X			
Freischalten (On-Demand) einer Fahrfunktion oder eines Motorprogramms			X		
Beheben eines Fehlers in der Airbag-Ansteuerung	X		X	X	X
Hinterlegen neuer gesetzlicher Rahmenbedingungen bzgl. automatisierter Fahrerassistenzsysteme				X	
Verbessern der Steuerung der Abgasnachbearbeitung	X			X	

Abbildung 2: Einstufung mittels Attribute

Um zu verdeutlichen wie die jeweiligen Attribute in Richtung Update-Ziel gedacht sind, ist als Beispiel eines solchen „einfachen“ Updates das Update einer Infotainment App oder die Softwarepflege zu betrachten. Für die Benutzung von Navigationskarten ist es so nicht notwendig, dass diese unmittelbar nach der Veröffentlichung eines Updates aktualisiert werden. So kann es auch dank fehlender Sicherheitsrisiken bei der Aktualisierung von Navigationsdaten nicht als Security-, Safety- oder Behörden-relevant eingestuft werden. Ein Update kann zwar Auswirkungen auf das Fahrerlebnis haben und somit relevant für Kunde und Hersteller sein, aber trotzdem kein Attribut zugewiesen bekommen.

Das Härten des Security-Systems ist als präventive Maßnahme nicht zeitkritisch oder imminent relevant für die Sicherheit von Verkehrsteilnehmern und bekommt somit nicht das Attribut Safety- oder Behörden-relevant. Als Schutz vor Fremdattacken ist es aber sehr wohl Security-relevant.

Das Schließen einer Sicherheitslücke hingegen, kann nach einer Attacke auf Systeme, die personenbezogene Daten oder Compliance-relevante enthalten, zeitnah erfolgen, um weiteres Datensammeln zu verhindern. Das ein solcher Angriff zwar sicherheitskritisch ist, aber kein Belang für die regulatorische Behörde darstellt, bekommt ein Update zur Schließung einer solchen Sicherheitslücke nur das Attribut Security-relevant. Das Attribut obligatorisch ergibt sich aus verschiedenen Zusammenhängen, aber in dem Fall gibt es keine gesetzlichen Verpflichtungen oder eine Verschiebung des Haftungsanspruches, die dieses Attribut erzwingen würden.

Die Inbetriebnahme einer Fahrfunktion oder eines neuen Motorprogrammes kann zu einer Gefährdung führen, wenn das spezifische Update oder die beinhaltete Software fehlerhaft ist. So ist ein solches Update Safety-relevant. Da es aber nicht um eine reaktive Schließung von Sicherheitslücken auf Softwareseite oder Ähnlichem geht, ergibt sich das Attribut Security-relevant nicht. Hersteller und Endnutzer können in verschiedenen Formen darüber entscheiden, ob ein solches Update überhaupt installiert werden soll (On Demand), dementsprechend sind die Attribute zeitkritisch, Behörden-kritisch, und obligatorisch nicht zweckgemäß.

Das Update zur Behebung eines Fehlers in der Airbag-Ansteuerung dagegen, bedarf der Kennzeichnung mit allen Attributen. Das fehlerhafte Zünden von Airbags stellt eine Gefährdung dar und ist damit Safety- und Behörden-relevant. Die Beseitigung dieses Fehlers ist dementsprechend auch zeitkritisch und obligatorisch.

Vor allem Behörden-relevant sind solche Updates, die zum Beispiel neue gesetzliche Rahmenbedingungen bezüglich automatisierter Fahrerassistenz auf dem Fahrzeug hinterlegen. Zusätzlich zu Behörden-relevant, greift das Attribut zeitkritisch dann in Fällen der Updates zur Verbesserung der Steuerung der Abgasnachbearbeitung.

2.6 Schlussfolgerung

Diese Art der Klassifizierung / Beschreibung von Updates entlang von Attributen wird in den nachfolgenden Kapiteln berücksichtigt und die Prozessschreibweise entsprechend ergänzt.

Das ursprüngliche Klassifizierungsschema hingegen war nicht geeignet, um die OTA-Prozesskette nachhaltig effizienter zu gestalten, indem in Abhängigkeit verschiedener Klassen die Anforderungen an die Aktivitäten des Prozesses abgestuft werden. Die drei dargestellten Dimensionen waren zu unabhängig voneinander, um eine Dimension auszuwählen. Auch eine Überarbeitung des Klassifizierungsschemas war nicht zielführend, da eine Harmonisierung der drei Dimensionen auf eine gemeinsame Basis nicht möglich war.

3 IDEALISIERTER OTA-PROZESS

In diesem Abschnitt wird ein idealisierter OTA-Updateprozess detailliert beschrieben. Die Beschreibung basiert auf einer Ist-Analyse anhand wissenschaftlicher Recherche, Einzelinterviews mit Vertretern der Automobilindustrie sowie einem im Rahmen des ersten Projektes durchgeführten Workshop am AQI mit benannten Experten der VDA-Mitglieder. Die Darstellung des idealisierten Prozesses soll als einheitliche Diskussionsgrundlage, für gemeinsame Kommunikation sowie weitere Analysen dienen. Zunächst werden die Grenzen des zu spezifizierenden Prozesses definiert, bevor die gewählte Prozessdarstellung beschrieben wird und der Prozess entlang seiner Aktivitäten detailliert beschrieben wird.

Für die Beschreibung dieses Prozesses wurde ein hoher Abstraktionsgrad gewählt, um die Beschreibung technischer und organisationsspezifischer Details zu vermeiden. Ziel war es, ein gemeinsames Verständnis zwischen allen Beteiligten aufbauen zu können. Jede beschriebene Aktivität kann durch weitere Aktivitäten vertieft und präzisiert werden. Bei steigender Präzision kann eine Allgemeingültigkeit nicht mehr gegeben sein, da die Unterschiede zwischen den verschiedenen Prozessen der beteiligten Mitgliedsfirmen nicht mehr berücksichtigt werden. Diese Details bzw. Unterschiede begründen sich auch mit dem Wettbewerb in der Industrie der noch ‚jungen' Technologie Over-The-Air Updates. (Auf technische Ausprägungen kann innerhalb dieses Dokumentes hingewiesen werden. Unter dem Gesichtspunkt des Wettbewerbs wird jedoch darauf verzichtet, diesbezüglich Anforderungen zu beschreiben oder zu definieren.)

3.1 Prozessgrenzen

Bei dem hier beschrieben Prozess geht es primär um die Verteilung von Software als OTA-Update auf die Fahrzeuge eines Fahrzeugherstellers. Nicht Gegenstand der Beschreibung ist die Erkennung des Updatebedarfs sowie die Herstellung und Bereitstellung des Software Updates.

Beginn des hier dargestellten Prozesses ist ein erkannter Software-Updatebedarf. Auf den detaillierten Prozess des Erkennens von Fehlern bzw. eines Software-Updatebedarfs wird in diesem Bericht nicht eingegangen. Für die Auslösung (*Start*) des hier dargestellten OTA-

Prozesses wird nur gefordert, dass ein solcher Software-Updatebedarf festgestellt worden ist. Die Identifikation eines solchen Bedarfs kann grundlegend über drei verschiedene Kanäle erfolgen:

(1) aus dem Fehlerabstellprozess

(2) durch die strategische Produktentwicklung

(3) durch Kunden-Feedback oder Kundenwunsch, z. B. nach einer Funktion im Fahrzeug[1]

Das notwendige Engineering des Update-Paketes selbst ist ebenfalls ein etablierter, stark wettbewerbsrelevanter Prozess, auf den in diesem Bericht nur verwiesen wird, um ihn durch OTA-spezifische Anforderungen ergänzen zu können.

Die Prozessbeschreibung *endet* nach der Rückmeldung über den Status der Installation des Updates durch das Fahrzeug an den OEM. Diese wird entweder als erfolgreich registriert (und ggf. als erfolgreich durchgeführter Rückruf der regulatorischen Behörde gemeldet) oder bei mehrfachem Fehlschlagen der Installation erfolgt die weitere Bearbeitung im Rahmen des konventionellen Werkstattprozesses. (Der Werkstattprozess ist nicht dezidierter Teil dieser Analyse.)

Fokus dieses Berichtes ist das OTA-Update im Sinne eines schreibenden Zugriffs auf das Fahrzeug. Vereinzelnd sind lesende Aktivitäten beschrieben, hierzu sei auf die weiteren Anforderungen beim Zugriff auf möglicherweise personenbezogene Daten hingewiesen, die ebenfalls nicht in diesem Bericht diskutiert werden.

Ferner kann bei Over-the-Air-Updates zwischen Konfigurationsupdates (COTA-Updates), Software- bzw. Firmware-Updates (sog. SOTA- bzw. FOTA-Updates) und dem Aktualisieren von Navigationskarten oder dem Schreiben von 3rd-Party-Content unterschieden werden. Im Folgenden wird auf alle Update-Typen gemeinsam als OTA-Update referenziert und mögliche Unterscheidungen über Attribute gesteuert (siehe Kapitel 2).

Der OTA-Prozess ist so gestaltet bzw. formuliert, dass jeweils nur ein Update behandelt wird, welches aber für verschiedene Zielsysteme ausgerollt werden kann. Das heißt, dass für mehrere Updates – auch wenn diese das gleiche Zielsystem ansprechen – entsprechend viele solcher Prozessketten parallel (zumindest gedanklich) instanziiert werden müssten.

[1] Die Vernetzung der Fahrzeuge ermöglicht die direkte Kommunikation mit dem Kunden. Unter den Nutzern gibt es Gruppen, die gerne Feedback an Hersteller zurückmelden und sich aktiv an Produktentwicklungen beteiligen.

3.2 Beschreibungsmittel für die Prozessdarstellung

Der idealisierte OTA-Prozess wurde sowohl grafisch modelliert als auch in Textform beschrieben. Eine genaue Erklärung und zu beachtende Aspekte bei der gewählten Darstellung werden in diesem Kapitel erläutert.

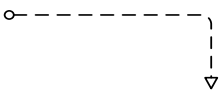
3.2.1 Modellierung mittels BPMN 2.0

Als Schreibweise für die Prozessdarstellung wurde BPMN 2.0 gewählt. Nachfolgende Legende beschreibt die verwendeten Elemente der Modellierungssprache:

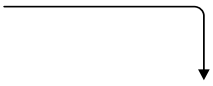
Auszuführende Tätigkeit

Optional: Ein Zahnrad-Paar weist auf einen automatisierten Dienst hin und eine Tabelle auf mögliche Geschäftsregeln, die zu berücksichtigen sind.

Optional: Ein umrandetes ‚S' weist darauf hin, dass die Aktivität relevant für die Simulation ist und dort berücksichtigt wurde.

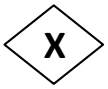


Informationsfluss: Beschreibt Informationsflüsse, die zur Kommunikation, zum Datenaustausch oder zur Informationsverteilung neben der Updateverteilung benötigt werden. (In Klammerausdrücken hinter der Bezeichnung des Informationsobjektes sind ggf. Bedingungen formuliert, unter denen diese Objekte ausgetauscht werden)

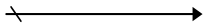


Kontrollfluss: Zeigt die Abfolge von Aktivitäten innerhalb einer Rolle.

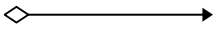
Und-Verknüpfung: Die darauffolgenden Aktivitäten sind alle auszuführen. Eine Reduzierung auf einen Pfad ist nicht erlaubt.



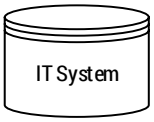
Entweder-Oder-Entscheidung: Der Prozess kann nur an einem Pfad fortgesetzt werden; die jeweiligen Bedingungen stehen an den Prozesspfaden.



Standard-Pfad nach einer Entweder-Oder-Entscheidung



Alternativer-Pfad in Abhängigkeit einer Bedingung nach einer Entweder-Oder-Entscheidung



IT-Systeme, wie Server, Datenbank-Systeme, Toolboxen, etc. (Wenn IT-Systeme mit der gleichen Bezeichnung mehrfach vermerkt sind, sind damit immer dieselben IT-Systeme gemeint.)

Datenobjekt, wie Dokument, Datei, Datenbank, etc.

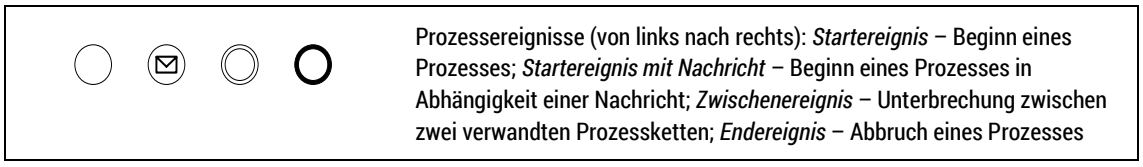


Abbildung 3: Legende der verwendeten BPMN 2.0

Alle Aktivitäten des Prozesses können durch eine eindeutige Prozess-ID identifiziert werden, auf die in dieser Dokumentation anschließend querverwiesen wird. Diese setzt sich aus drei Bestandteilen zusammen:

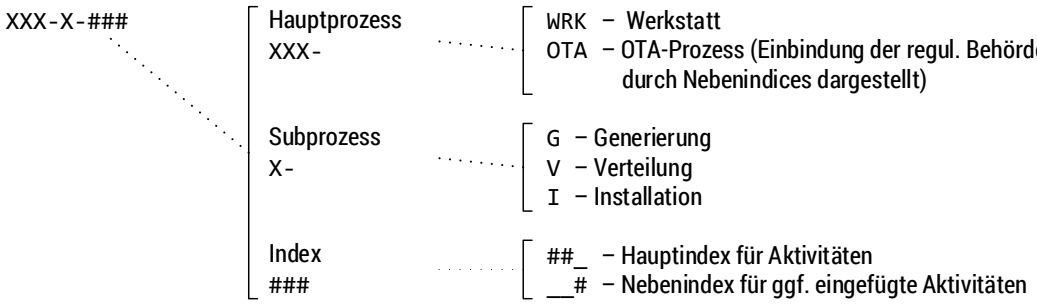


Abbildung 4: Definition der Prozess-ID

Während der Ist-Aufnahme wurden zwei Hauptprozesse aufgenommen: der konventionelle Update-Prozess (WRK) für Software-Updates, bei dem die Updates durch Tester in Werkstätten durchgeführt werden, und der Prozess, um solche Updates Over-The-Air (OTA) durchzuführen.

Aktivitäten, die auch Teil des konventionellen Prozesses sein könnten, sind in **Schwarz** dargestellt sofern sich diese auch im OTA-Prozess wiederfinden. Alle neuen oder durch OTA-Updates veränderten Aktivitäten sind in **Blau** dargestellt.

Beide Prozesse variieren, wenn eine regulatorische Behörde eingebunden werden muss. Diese Alternationen des Prozesses sind in **Rot** hervorgehoben. Die Aktivitäten und Informationsflüsse entfallen entsprechend, wenn die regulatorische Behörde nicht in den Updateprozess involviert ist. In der Textvariante wird mit dem Hinweis „nur regul. Behörde“ auf die Erweiterung der Prozesskette hingewiesen.

Abbildung 5 zeigt ein entsprechendes Beispiel.

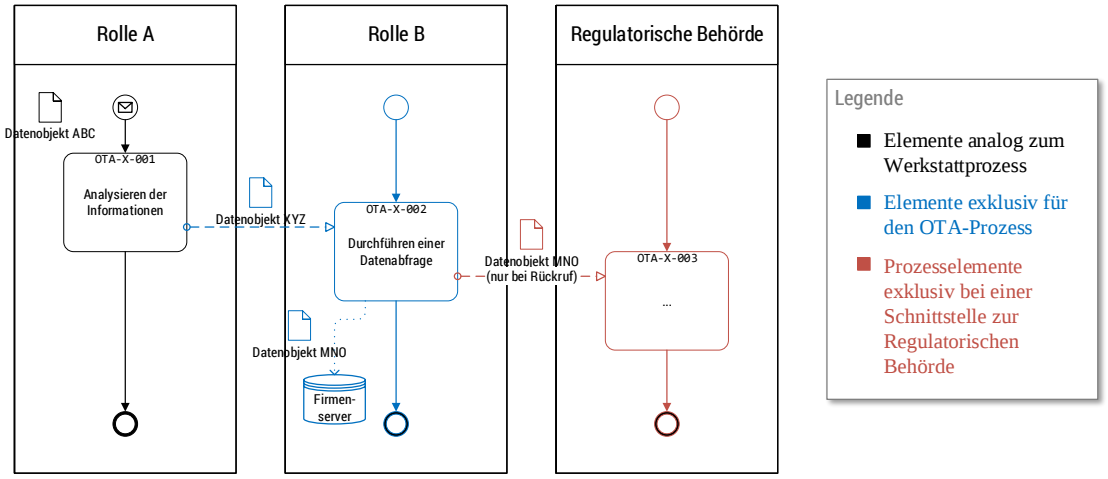


Abbildung 5: Beispiel der Darstellung von Prozessvariationen

3.2.2 Textbeschreibung

Jede Aktivität ist zusätzlich tabellarisch dargestellt. Dabei beschreibt jede Zelle der Tabelle eine der folgenden Eigenschaften einer Aktivität. Diese werden allerdings nur aufgeführt, wenn sie für diese Aktivität relevant sind:

- **Input**: Informationsobjekt, das für die Aktivität benötigt wird.
- **Kurz-Beschreibung**: in der Beschreibung der Aktivität ist in ***kursiv und fett*** der jeweilige Verantwortliche für die Aktivität hervorgehoben und nur in *kursiv* dargestellt sind alle weiteren Rollen, die erwähnt werden.
- **RACI-Modell**: Das RACI-Modell beschreibt die organisatorischen Verhältnisse beteiligter Rollen einer Aktivität zueinander.
  Responsible: verantwortliche Rolle für die Durchführung.
  Accountable: Rolle, die ggf. die rechenschaftspflichtige Gesamtverantwortung trägt.
  Consulted: Rolle, die beratend die Durchführung unterstützt.
  Informed: Rolle, die während der Durchführung der Aktivität informiert werden muss.[2]
- **Output**: Informationsobjekt, das aus der Aktivität entsteht.
- **Konditionen**: mögliche verschiedene Ausprägungen, die in Abhängigkeiten zu anderen Faktoren, z. B. den Attributen des Updates stehen.
- **Chancen**: Chancen haben einen positiven Einfluss auf einen der Stakeholder des OTA-Prozesses.

[2] Das RACI-Modell ist beispielhaft eingefügt worden, um die Komplexität und verschiedenen Schnittstellen innerhalb der jeweiligen Organisationen zu betonen. Die Unternehmen selbst sollten das RACI-Modell entsprechend ihrer eigenen Strukturen überarbeiten.

- **Risiken**: Risiken haben einen negativen Einfluss auf einen der Stakeholder des OTA-Prozesses.
- **Security-Risiken**: Security-Risiken sind Risiken, die auf eine Cyber-Security Schwachstelle zurückgeführt werden können. Diese können durch Dritte zum Schaden eines Stakeholders des OTA-Prozesses ausgenutzt werden.
- **Parameter**: beschreibt, die für die Simulation relevanten Einflussparameter für die Aktivität und deren Auswirkung auf die Prozesskette.
- **UN ECE Anforderung**: sofern sich aus den Regularien der UN ECE Anforderung an eine der Aktivitäten ergeben, ist diese hier im Original Wortlaut mit Verweis auf das Ursprungskapitel hinterlegt.

Die gesamte Tabelle ist exemplarisch wie folgt aufgebaut:

Input: Informationsobjekt ABC
Rolle A analysiert das Datenobjekt ABC und informiert *Rolle B* mittel Datenobjekt XYZ.
R Rolle A A Update-Koordinator C (Halter) I Rolle B
Output: Informationsobjekt XYZ
Konditionen: Wenn zeitkritisch: a) dann … b) dann … Wenn Safety-relevant: a) dann …
Chance-E###: ... **Risiko-E###**: … **Security-Risiko-E###**: …
Parameter1: Beschreibung des Parameters Parameter2: …
UNE ECE REQUIREMENTS[3]: #.#.# …

Tabelle 1: Beispiel einer Textbeschreibung von Prozess-Aktivitäten

[3] Die Verweise beziehen sich immer auf den Annex A des ‚Final Draft' der UN ECE Recommendation vom 21.09.18.

Im Vergleich zum im ersten Projekt entworfenen Prozess wurden alle Aktivitäten überarbeitet. Wurden Aktivitäten deutlich in ihrem Inhalt verändert, wurde die ursprüngliche Aktivität entfernt und die neue Aktivität mit einem um „1“ höheren Index versehen. Alle entfernten Aktivitäten finden sich im Anhang. Eine entsprechende Begründung, warum sie entfallen sind, ist ergänzt worden. Alle vollständig neuen Aktivitäten sind mit [neu] in der Überschrift hervorgehoben, alle optionalen Aktivitäten entsprechend mit [optional].

3.2.3 Rollen im Prozess

Im Prozess werden verschiedene Rollen stellvertretend für den Bearbeiter, die Abteilung oder die Organisationseinheit in der jeweiligen spezifischen Lieferkette verwendet. Die beschriebenen Rollen können in ihrer Bezeichnung aber auch ihrem Aufgabenbereich in den verschiedenen Unternehmen variieren oder sogar in Personalunion ausgeführt werden. Sie dienen hier zur Strukturierung der Inhalte:

- Der *Update-Koordinator* übernimmt auf Seiten des OEM die Koordination des Software-Updates und repräsentiert den gesamtverantwortlichen Entscheider für den Erfolg des Updates.
- Der *Entwicklungsverantwortliche* verantwortet die technische Lösung.
- Der *Entwickler* setzt die Lösung um.
- Der *Digitale Dienstleister* repräsentiert die interne oder externe Abteilung / Firma, die die bei OTA hinzukommenden Aufgaben der Verteilung des Updates übernimmt. Das heißt, die ‚Lücke‘ in der Übertragung schließt, die zwischen den Servern des OEM und dem lokalen Speicher im Fahrzeug existiert.

Darüber hinaus existieren folgende Rollen:

- *Halter*
- *Fahrzeug*
- *Regulatorische Behörde*

3.3 UN ECE Task Force on OTA-Updates

Eines der aktuell treibenden gesetzlichen Vorhaben basiert auf den Ergebnissen der „UN ECE Task Force on Cyber Security and Over-the-Air Issues“, die zukünftig in neue gesetzliche Anforderungen an OTA-Updates überführt werden können. Die Task Force ist eine Untergruppe der „Informal Working Group on Intelligent Transport Systems / Automated Driving (IWG

IST/AD)“ des „UNECE World Forum for Harmonization of Vehicle Regulations (WP.29)”. Aufgrund dieser Relevanz sind die Anforderungen der Task Force in den folgenden idealisierten OTA-Prozess eingearbeitet und bei den relevanten OTA-Aktivitäten verortet (vgl. Annex A der Recommendation für Softwareupdates).

Innerhalb der UN ECE wurden letztlich zwei Recommendations für die beiden Bereiche Cybersecurity und Software Updates verfasst. Dabei ist zu betonen, dass im Bereich der Updates der Aufgabenbereich im Laufe der Bearbeitung nicht mehr nur OTA-Updates umfasste, sondern generell auf Softwareupdates erweitert wurde.

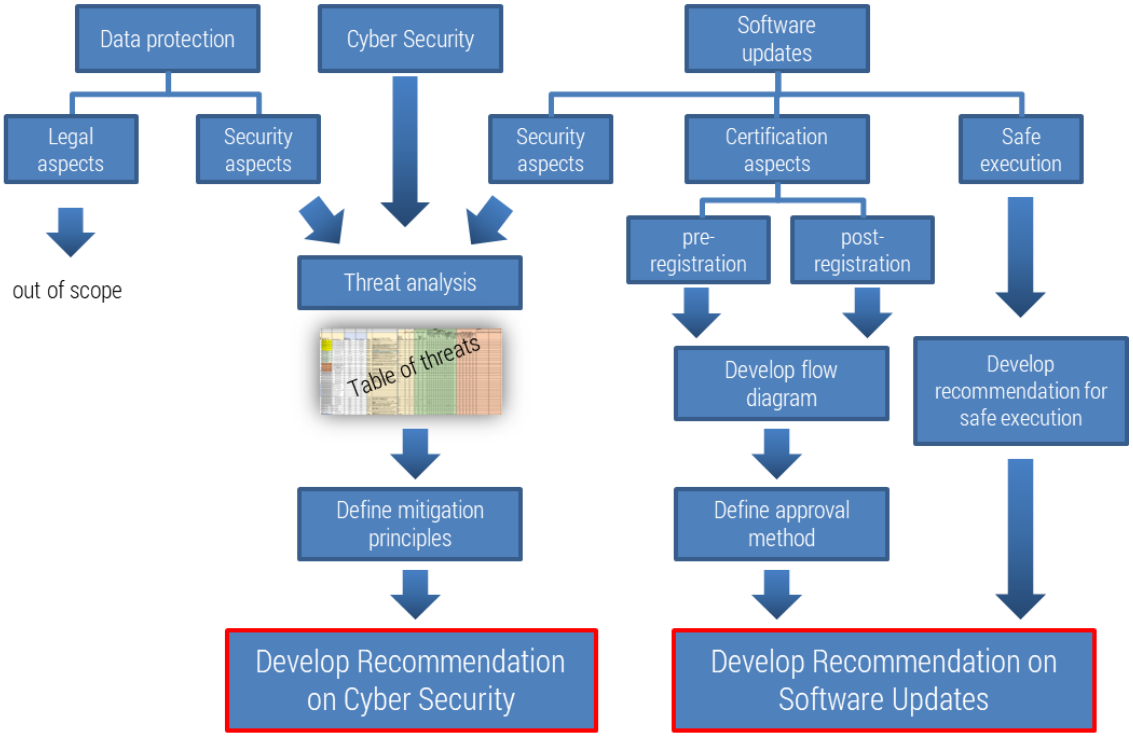


Abbildung 6: Scope of Works der UN ECE TF on CS/OTA

Abbildung 7 zeigt die Inhalte, die in der Recommendation für die Softwareupdates erarbeitet wurden.

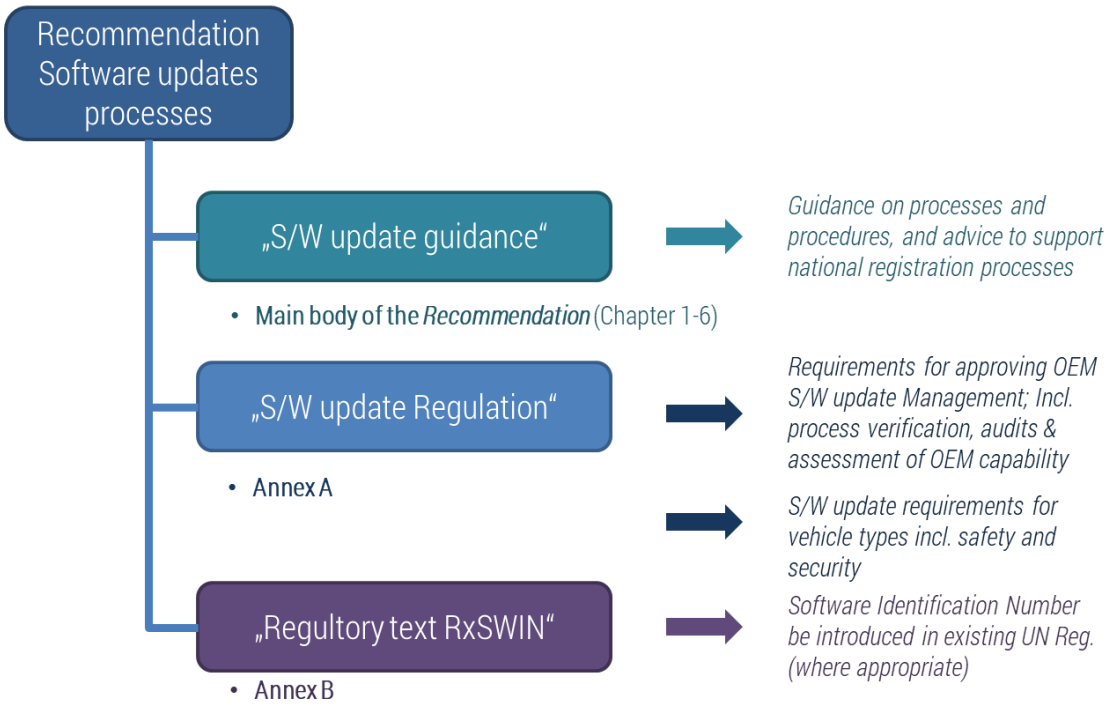


Abbildung 7: Inhalte der Recommendation on Softwareupdates

Im Hauptteil der „S/W update guidance“ sind Anpassungen an den Zertifizierungsprozess von Fahrzeugen beschrieben, um die behördlichen Anforderungen aus der UN ECE an Softwareupdates zu berücksichtigen. Dabei sind neben den Zertifizierungsaspekten Aspekte der Safety und Security zu beachten, um Softwareupdates rechtssicher durchführen zu können.

Hierbei wird vor allem die Beziehung zwischen OEM und regulatorischer Behörde beschrieben, während im Annex A „S/W update Regulation“ konkret Anforderungen an den OTA-Prozess und die organisatorischen Voraussetzungen selbst gestellt werden. Diese Anforderungen aus dem Annex A werden nachfolgend in den idealisierten Prozess integriert und den entsprechenden Aktivitäten zugeordnet. Dabei wird immer der englische Originaltext verwendet.[4]

Darüber hinaus werden jedoch noch allgemeine Prozessanforderungen bzw. Prozessvoraussetzungen im Annex A beschrieben.

So bedarf es grundsätzlich einer Zertifizierung eines Software Update Management Systems (SUMS), um zu bestätigen, dass die Anforderungen der UN ECE eingehalten werden. Mit einer solchen Bestätigung erteilt die regulatorische Behörde erst eine Typgenehmigung zur Aktualisierung von Software für entsprechende Fahrzeugtypen. Ein solches Zertifikat ist für drei Jahre gültig und bedarf anschließend einer erneuten Zertifizierung. Ggf. kann es sogar zwischendurch schon zu einer erneuten Überprüfung des SUMS Zertifikats kommen. Erlischt das

[4] Im Dokument wird auf den Stand vom 21. September der UN ECE Dokumentation referenziert – „Final Draft“: https://wiki.unece.org/pages/viewpage.action?pageId=60362218

SUMS Zertifikat aufgrund einer fehlenden Verlängerung, Nicht-Anzeige von Änderungen oder einer negativen Zwischenprüfung hat dies jedoch kein Einfluss auf bereits erteilte Fahrzeug-Typgenehmigungen.[5]

Außerdem konnten folgende Anforderungen keiner spezifischen Aktivität zugeordnet werden.

Darüber gibt es eine Reihe von Anforderungen der UN ECE, die keiner spezifischen Aktivität im nachfolgenden idealisierten Prozess zugeordnet werden konnten. Diese werden aus Gründen der Transparenz nachfolgend dokumentiert.

- 7.1.1.1 A process whereby information relevant to this regulation is documented and securely held at the vehicle manufacturer and can be made available to an Approval Authority or Technical Service upon request without any burden;
- 7.1.1.2 A process whereby information regarding all initial and updated software versions, including integrity validation data, and relevant hardware components of a type approved system can be uniquely identified;
- 7.1.1.3 A process whereby, for a vehicle type that has an RXSWIN, information regarding the RXSWIN of the vehicle type before and after an update can be accessed and updated. This shall include the ability to update information regarding the software versions and their integrity validation data of all relevant software for each RXSWIN.
- 7.1.1.4 A process whereby, for a vehicle type that has an RXSWIN, the vehicle manufacturer can verify that the software version(s) present on a component of a type approved system are consistent with those defined by the relevant RXSWIN;
- 7.1.2.1 Documentation describing the processes used by the vehicle manufacturer for providing software updates and any relevant standards used to demonstrate their compliance;
- 7.2.1.2.2 The RXSWIN shall be easily readable in a standardized way via the use of an electronic communication interface, at least by the standard interface (OBD port).
- 7.2.1.2.3 The vehicle manufacturer shall protect the RXSWINs on a vehicle against unauthorised modification. At the time of Type Approval, the means implemented to protect against unauthorized modification of the RXSWIN chosen by the vehicle manufacturer shall be confidentially outlined.

Im dritten Dokument, dem Annex B “Regultory text RXSWIN” wird die Einführung einer „Regulation X Softwareidentifikationsnummer“ beschrieben, deren Thematisierung jedoch außerhalb des inhaltlichen Rahmens dieses Berichtes liegt.

[5] In diesem Absatz sind die Kapitel 3 bis 6 für einen Überblick zusammengefasst. An dieser Stelle sei auch auf die Kapitel 9 „Conformity of Production“ und Kapitel 10 „Penalties for non-conformity production“ hingewiesen.

3.4 Prozessinhalte

Der Prozess eines Updates gliedert sich grundlegend in drei Unterprozesse, die von zwei zentralen Aufgaben begleitet werden.

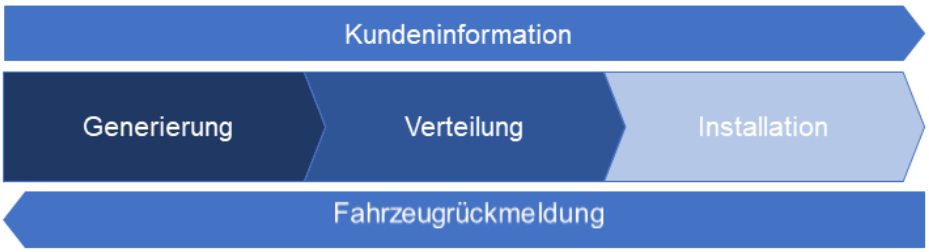


Abbildung 8: Schema des Updateprozesses

Zur **Generierung** zählen Aufgaben wie die Klassifizierung und das Kommunizieren des Updatebedarfs, ggf. die Einbindung der regulatorischen Behörde, die Entwicklung des Updates sowie die Qualitätskontrolle dieses Updates und ein anschließendes Veröffentlichen des Updates unter entsprechendem Schutz durch Cyber-Security Maßnahmen.

Die **Verteilung** des Updates beginnt mit dem Bereitstellen des Updates in einem Distributionsnetzwerks, setzt sich fort über das Benachrichtigen des Fahrzeuges und Informieren des Kunden, der das Update genehmigen muss, und endet mit der Übertragung des Updates zum Fahrzeug.

Anschließend beginnt die **Installation** zunächst mit einer Verifikation des Updates und der Überprüfung der Fahrzeugkonfiguration bevor die Installation angestoßen wird. Nach der Durchführung der Installation ist diese zu testen und das Ergebnis zurückzumelden.

Entlang der gesamten Prozesskette ist stets die **Kundeninformation** zu berücksichtigen. Kontinuierlich sind die Kunden über Updateverfügbarkeit, -inhalte und -fortschritte zu informieren. Eine begleitende **Fahrzeugrückmeldung** unterstützt das Identifizieren und Vermeiden von Problemen und ihre anschließende Behebung für sämtliche betroffene Fahrzeuge.

Im Folgenden wird nun der OTA-Updateprozess beschrieben, dargestellt und untersucht. Jeweils für die Prozessschritte Generierung, Verteilung und Installation.

3.5 Update-Generierung

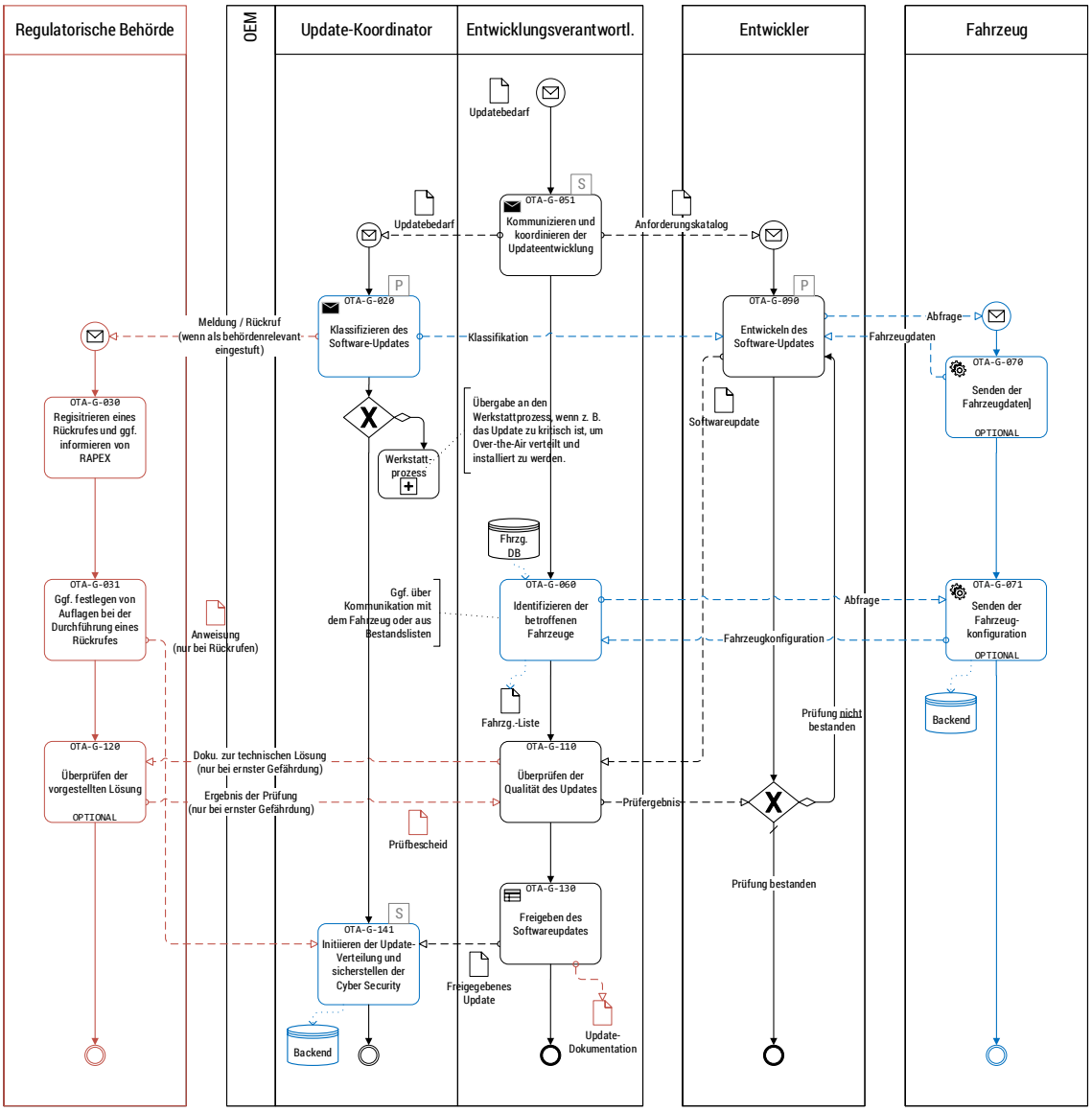


Abbildung 9: Prozessmodellierung "Generierung"

OTA-G-051 [neu]

Input: Updatebedarf, Anforderungskatalog / Lastenheft
Der ***Entwicklungsverantwortliche*** koordiniert die Entwicklung des Updates mit den *Entwicklern* innerhalb und / oder außerhalb des OEMs. Hierzu formuliert und stimmt er die Anforderungen an das Update ab[6]. Außerdem kommuniziert er den Start der Update-Entwicklung an den *Update-Koordinator*.

[6] Je nach Entwicklungsmethode ist dies ein kontinuierlicher Prozess entlang der Update-Entwicklung oder eine vorgelagerte Aktivität, die mit einem Ergebnis, z. B. einem Lastenheft, abschließt.

R	Update-Koordinator
A	Update-Koordinator
C	Entwicklungsverantwortlicher
I	Entwickler, regulatorische Behörde

Output: Updatebedarf, Anforderungskatalog / Lastenheft, Update-Dokumentation analog UN ECE

`duraGenerateUpdate`: Dauer [in Stunden], die benötigt wird, um das Update zu generieren. Bezieht sich auf die Dauer der Aktivitäten aus dem Subprozess Generierung einschließlich der Aktivität OTA-G-130.

`nVehicles`: Anzahl der Fahrzeuge [-], die das Update benötigen. Eine Verfeinerung und Überprüfung der Kennzahl folgt in OTA-G-060.

UN ECE REQUIREMENT:

7.1.2.5. Documentation for all software updates for that vehicle type describing:

1. The purpose of the update;
2. What systems or functions of the vehicle the update may impact;
3. Which of these are type approved (if any);
4. If applicable, whether the software update affects any of the relevant requirements of those type approved system;
5. Whether the software update affects any system type approval parameter;
6. Whether an approval for the update was sought from an approval body;
7. How the update may be executed and under what conditions;
8. Verification that the software update will be conducted safely and securely.
9. Verification that the software update has undergone adequate verification and validation procedures.

Erläuterung

Durch die Verteilung von Softwareupdates Over-the-Air ergeben sich auch neue Anforderungen, die in das Lastenheft ergänzt werden müssen. Diese ergeben sich einerseits aus dem Stand der Technik, wie z. B. Anforderungen an die Übertragung oder die Installationsroutine in Abhängigkeit der Leistung der ECU, andererseits aus den Anforderungen der UN ECE und dem eigenen Ziel einen robusten OTA-Updateprozess zu etablieren. Hierzu zählen insbesondere Anforderungen der Safety, die entsprechend des Ziels des Updates zu definieren sind, z. B. dass das Fahrzeug nicht am Hang geparkt sein darf, wenn ein Update für das Bremssystem installiert wird, oder dass ein Fahrzeug nicht während der Fahrt aktualisiert werden darf.

Der zitierte Paragraph der UN ECE Regularien dient einer ersten Einschätzung, was seitens der UN ECE gefordert wird. Diese Anforderungen implizieren auch weitere Anforderungen, durch die das Lastenheft ergänzt werden könnte. Weitere Implikationen ergeben sich aus den weiteren hier beschriebenen Aktivitäten.

Die durch die UN ECE geforderte Dokumentation kann als Entwurf dem Updatebedarf ergänzt werden, um so auch in OTA-G-020 eine bessere Klassifizierung und damit ggf. Einbindung der

regulatorischen Behörde vornehmen zu können. Das finale Dokumentation könnte während OTA-G-130 „Freigeben des Softwareupdates“ als Anlage zum Update-Paket an die regulatorische Behörde weitergeleitet werden.

OTA-G-020

Input: Updatebedarfsmeldung, Anforderungsbedarf / Lastenheft, Dokumentation analog UN ECE
Der ***Update-Koordinator*** klassifiziert das Software-Update.[7]
R Update-Koordinator A Update-Koordinator C Entwicklungsverantwortlicher I Entwickler, regulatorische Behörde
Output: Ergänzung der Updatebedarfsmeldung um Updateklasse; ggf. Meldung an die *Regulatorische Behörde*
Konditionen: **(A)** Wenn das Update produktsicherheitsrelevant ist, ist die *Regulatorische Behörde* einzubinden (→ OTA-G-030). **(B)** Wenn das Update zu kritisch ist, um Over-The-Air verteilt zu werden, folgt hier der Abbruch und die Übergabe an den Werkstattprozess (→ WRK-G-040) **(C)** Wenn das Update zeitkritisch ist, darf eventuell die regulatorische Behörde nachrangig eingebunden und erst ist Anschluss über die Maßnahme informiert werden.
Chance-E001: Durch das Klassifizieren von OTA-Updates und einer damit einhergehenden Variation der Prozessdurchführung in Abhängigkeit der Klassifikation, kann der Update-Prozess effizienter gestaltet werden. **Risiko-E002**: Eine falsche Klassifizierung kann zu einer Verletzung einer Meldepflicht bei der *Regulatorischen Behörde* führen. **Risiko-E003**: Eine falsche Klassifizierung kann dazu führen, dass das Update nicht ausreichend getestet wurde und es in Folge dessen zu Fehlern im Feld kommt. **Risiko-E027:** Teilweise existieren gesetzliche Regelungen bzgl. der Verantwortung zur Wartung von Fahrzeugen für den Verkäufer und somit liegt diese Verantwortung eventuell bei der Werkstatt des Händlers und nicht beim OEM, so dass der OEM selbst keine Updates aufspiele darf.
`Class`: diese Klasse speichert die Updateeigenschaften entsprechend der vorgenommenen Klassifizierung.
UN ECE REQUIREMENT: 7.1.1.8 A process to assess, identify and record whether a software update will affect any type approved systems. This shall consider whether the update will impact or alter any of the parameters used to define the systems the update may affect or whether it may change any of the parameters used to type approve those system (as defined in the relevant legislation); 7.1.1.9 A process to assess, identify and record whether a software update will add, alter or enable any functions that were not present, or enabled, when the vehicle was type approved or alter or disable any

[7] Klassifizierungsschemata vgl. Kapitel 2.4

other parameters or functions that are defined within legislation. The assessment shall include consideration of whether: 1. Entries in the information package will need to be modified 2. Test results no longer cover the vehicle after modification 7.1.1.10 A process to assess, identify and record if a software update will affect any other system required for the safe and continued operation of the vehicle or if the update will add or alter functionality of the vehicle compared to when it was registered 7.1.4.2 The vehicle manufacturer shall demonstrate the processes and procedures they will use to ensure that, when an over the air update requires a skilled person, such as a mechanic, in order to complete the update process, the update can only proceed when such a person is present.

Erläuterung

Wenn der OEM erkennt, dass durch ein Softwareupdate eine Rückwirkung auf die Typgenehmigung durch das Softwareupdate resultiert, muss der Prozess mit der regulatorischen Behörde angestoßen werden. Es ist zu prüfen, ob es einer Erweiterung der bestehenden Typgenehmigung bedarf oder eine neue Typgenehmigung benötigt wird.

OTA-G-030 [nur reg. Beh.]

Input: Meldung / Rückruf-Anmeldung	
Die ***Regulatorische Behörde*** registriert den Rückruf und informiert ggf. RAPEX bzgl. des Rückrufs.	
R	Regulatorische Behörde
A	Update-Koordinator
C	Update-Koordinator
I	ggf. RAPEX
Output: Rückrufregistrierung	

OTA-G-031 [nur reg. Beh.]

Input: Rückrufregistrierung	
Ggf. wird durch die ***Regulatorische Behörde*** eine Frist bis zur Erfüllung des Rückrufs und/oder weitere Auflagen festgesetzt.	
R	Regulatorische Behörde
A	
C	
I	Update-Koordinator
Output: ggf. Anweisung an OEM	

UN ECE REQUIREMENT: 8.1. Every modification of the vehicle type shall be notified to the approval authority which granted the approval. The approval authority may then either: 8.1.1. Consider that the modifications made are unlikely to have an appreciable adverse effect and that in any case the vehicle still complies with the requirements; or 8.1.2. Require a further test report from the technical service responsible for conducting the tests.

OTA-G-090

Input: Lastenheft, Fahrzeugdaten
Der ***Entwickler*** entwickelt das neue Software-Update. Dazu zählt es auch technische Randbedingungen für das Update zu definieren. Ggf. können vorhandene Fahrzeugdaten genutzt werden, um die Diagnose und das Update selbst zu verbessern (→ OTA-G-070). Hierzu kann der *Entwickler* optional eine entsprechende Anfrage an das *Fahrzeug* senden.
R Entwickler A Entwicklungsverantwortlicher C (ggf. Fahrzeug), (ggf. Abteilung, die den Updatebedarf gemeldet hat) I (ggf. Update Koordinator, falls es relevante Änderungen gibt)
Output: Softwareupdate, ggf. Abfrage nach Fahrzeugdaten
Konditionen: **(A)** Wenn zeitkritisch, dann sollte das Update getrennt von weiteren Paketen verteilt werden.
Chance-E004: Der Fernzugriff auf Daten zur Unterstützung der Fehlerdiagnose und damit der Fehlerbehebung kann zu einer verbesserten Fehler-Interpretation und Lösungsfindung führen. **Risiko-E005:** Die Rahmenbedingungen für den Datenaustausch personenbezogener Daten zwischen OEM und möglichem externen *Entwickler* sind zu klären. **Risiko-E006:** Das entwickelte Softwareupdate entspricht nicht den Vorgaben für sichere Software und macht das aktualisierte Fahrzeug anfällig für Angriffe (z. B. Datendiebstahl, IP-Diebstahl, Malware, etc.) **Security-Risiko-E007:** Mangelhaftes Key Management während der Softwareentwicklung (aber auch während des gesamten Lifecycles) gibt Angreifern die Möglichkeit Authentifizierungsschlüssel zu zerstören, auszutauschen, oder deren Verfügbarkeit einzuschränken. **Security-Risiko-E007b**: Die Freigabe des Updates zur Verteilung über USB-Sticks / Smartphone öffnet eine weitere Schnittstelle deren Absicherung im gesamten Design berücksichtigt werden muss. **Chance-E026:** Bei sicherheitskritischen Updates kann eine getrennte Verteilung der Updates, die das Verhalten, das Interface oder andere sonstige kunden-wahrnehmbare Features verändern, hilfreich sein, um eine schnelle Genehmigung des Updates zu erhalten.
`sizeUpdate`: Größe des Updates [in MB] `speed#G`: Geschwindigkeit verschiedener Übertragungstechniken (WiFi, 3G, 4G und 5G oder via USB-Stick) `p#G`: Verteilung, welche Übertragungstechnik voraussichtlich zu welchem Anteil in Prozent von 100% für die Übertragung ins Fahrzeug genutzt wird.

UN ECE REQUIREMENT:

7.1.1.5. A process whereby any interdependencies of the updated system with other systems can be identified;

7.1.2.2. Documentation describing the configuration of any relevant type approved systems before and after an update, this shall include unique identifiers for the type approved system's hardware and software and any relevant vehicle or system parameters;

7.1.2.3. For every RXSWIN, there shall be documentation describing the software relevant to the RXSWIN of the vehicle type before and after an update. This shall include information of the software versions and their integrity validation data for all relevant software for each RXSWIN.

7.1.3.3. The processes used to verify and validate software functionality and code for the software used in the vehicle are appropriate.

Erläuterung

Bei der Entwicklung des Updates sind einerseits grundlegende Cyber-Security-relevante Maßnahmen zu berücksichtigen (wie Secure Coding), andererseits Maßnahmen, um das Update-Paket für eine Übertragung Over-the-Air zu optimieren (wie Berücksichtigung relevanter Frameworks). Dazu zählen auch Konzepte, wie zum Beispiel, die Übertragung nur einzelner Differenzinformationen (z. B. Diff-Files), also solcher Datenpakete, die nur den veränderten Code (und entsprechende Skripte zum Schutz und zur Installation) enthalten, oder das Streaming von Update-Inhalten. Dies ist entsprechend im Design zu berücksichtigen.

Die Optimierung für verschiedene Übertragungstechniken ist abhängig sowohl von der Kritikalität als auch von der Benutzergruppe. Kritische Updates könnten bevorzugt über Mobilfunk verteilt werden. Hierdurch steigt die Netzabdeckung deutlich an und die Wahrscheinlichkeit für einen Updateabbruch (→ siehe OTA-G-130) sinkt. Das Update kann so schneller das Zielfahrzeug erreichen – auch bei langsamerer Geschwindigkeit, da die Erreichbarkeit höher ist. Ggf. muss jedoch der OEM die Kosten für die Übertragung selbst tragen.

Bzgl. der Kundengruppe ist einerseits deren Affinität zur Technik zu berücksichtigen, damit auch die Sensitivität gegenüber notwendigen Updates, andererseits auch die jeweilige Kaufkraft und damit mögliche Unterschiede in den verbauten Connectivity-Modulen bzw. abgeschlossenen Mobilfunk-Paketen in den Zielfahrzeugen.

Optional kann das Softwareupdate entweder auf einen mobilen Datenspeicher heruntergeladen werden (z. B. ein Smartphone oder ein USB-Stick) und von diesem Datenträger direkt in das Fahrzeug eingespielt werden oder über ein Smartphone als Funkmodul mittels Tethering ohne direkte Fahrzeugkonnektivität heruntergeladen werden. Diese Übertragungsvarianten stellen allerdings ein weiteres Cyber-Security-Risiko dar, weil eine weitere Schnittstelle geöffnet wird und fremde Geräte genutzt werden, die nicht durch den OEM abgesichert werden können.

Bei einer Entwicklung des Updates innerhalb der Lieferkette muss eine Absicherung in der gesamten Kette gegeben sein und das Update-Paket gegen Manipulationen von außen geschützt werden.

Bzgl. Chance-E026: die getrennte, also atomare Verteilung von Updates ermöglicht es z. B., dass Security-kritische schnell an die Zielfahrzeuge verteilt werden und auch genehmigt werden, da ein atomares Security-Update geringere Anforderungen an die Installation haben, als andere Updates. Eine geringe Zeit, die das Fahrzeug nicht benutzbar ist, motiviert den Fahrer eher das kritische Fahrzeug aufzuspielen, als wenn die Aktualisierung Teil eines größeren Paketes ist, für das das Fahrzeug entsprechend länger nicht genutzt werden kann. Nachteil einer atomaren Verteilung könnte es jedoch sein, dass den Kunden zu häufig Anfragen nach Updates erreichen. Dieser könnte sich durch diese Häufigkeit gestört fühlen und einen Automatismus zum Aufschieben bzw. Ablehnen dieser Updates entwickeln.

OTA-G-070 [optional]

Input: Abfrage nach Fahrzeugdaten	
Das ***Fahrzeug*** sendet – falls eine Abfrage in OTA-G-090 initiiert wurde – die Fahrzeugdaten an den *Entwickler*. Zu den Fahrzeugdaten zählen dabei z. B. Diagnose-Daten oder je nach Zustimmung der Halter auch Belastungs- oder Fahrprofile. Mit Hilfe dieser Daten kann die Updateentwicklung verbessert werden.	
R	Fahrzeug
A	Entwicklungsverantwortlicher
C	(ggf. Halter / Nutzer für Einverständnis zur Datennutzung)
I	Entwickler
Output: Fahrzeugkonfiguration / Fahrzeugdaten (an OTA-G-090);	
Risiko-E008: Bei unklaren Rahmenbedingungen, unter denen Daten aus den Fahrzeugen abgerufen werden dürfen, könnten ggf. regulatorische Regelungen verletzt werden. **Security-Risiko-E009**: Unautorisierter Datenzugriff auf Unternehmens- oder personenbezogene Daten bei der Datenübermittlung. **Security-Risiko-E010**: Die Datenschnittstelle und/oder andere Fahrzeugelektronik werden manipuliert oder missbraucht.	

Hinweis zur Simulation

Diese Aktivität wurde in der Simulation (Kapitel 4) nicht berücksichtigt, da die Trigger-Aktivität OTA-G-090 (idealerweise) nicht beendet ist, bevor das gewünscht Feedback vom Fahrzeug berücksichtigt werden kann. Damit ist OTA-G-090 zeitlich ausschlaggebend für die Simulation.

OTA-G-060

Input: Fahrzeugkonfiguration (aus OTA-G-071); Daten aus eigener Datenbank
Der ***Entwicklungsverantwortliche*** beginnt die betroffenen Fahrzeuge zu identifizieren. Dies kann durch direkte Abfrage der OTA-fähigen Fahrzeuge erfolgen (‚Shoulder-Tap') oder indirekt durch Zugriff auf

eine Datenbank, in welche die Fahrzeuge regelmäßig (in einem festgelegten Zeitintervall) ihre Konfiguration / weitere Eigenschaften schreiben könnten (‚Fhrzg.-DB'). Ggf. wird diese Datenbank auch mit weiteren Daten aus unternehmensinternen Quellen ergänzt (z. B. Kundenportal).
R Entwicklungsverantwortliche A Entwicklungsverantwortliche C I Fahrzeuge
Output: Liste mit den zu aktualisierenden Fahrzeugen; ggf. Abfrage (an OTA-G-071)
Chance-E011: Durch den Fernzugriff auf die Fahrzeuge können die betroffenen Fahrzeuge zuverlässiger identifiziert und ihr aktueller Softwarestand analysiert werden. So kann vermieden werden, dass ein Update-Vorgang initiiert wird, der auf Grund fehlender Kompatibilität fehlschlagen könnte.
UN ECE REQUIREMENT: 7.1.1.5. A process whereby any interdependencies of the updated system with other systems can be identified; 7.1.1.6. A process whereby the vehicle manufacturer can identify target vehicles for a software update; 7.1.2.4. Documentation listing target vehicles for the update and verification of the compatibility of the registered configuration or last known configuration of those vehicles with the update.

OTA-G-071 [optional] [neu]

Input: Abfrage
Das ***Fahrzeug*** sendet entsprechend OTA-G-060 die Fahrzeugkonfiguration (Softwareversionen, verbaute Hardware, etc.).
R Fahrzeug A Entwicklungsverantwortlicher C I
Output: Fahrzeugkonfiguration (an OTA-G-060)
Risiko-E-012: Bei unklaren Rahmenbedingungen, unter denen Daten aus den Fahrzeugen abgerufen werden dürfen, könnten ggf. regulatorische Regelungen verletzt werden. **Security-Risiko-E013:** Unautorisierter Datenzugriff auf Unternehmens- oder personenbezogene Daten bei der Datenübermittlung. **Security-Risiko-E014**: Die Datenschnittstelle und/oder andere Fahrzeugelektronik werden manipuliert oder missbraucht.

OTA-G-110

Input: Softwareupdate; Dokumentation über das Update
Der ***Entwicklungsverantwortliche*** überprüft die Qualität des neuen Softwarestandes. Bei Verdacht auf ernste Gefährdung[8] wird die gewählte Lösung dokumentiert und an die *Regulatorische Behörde* zur weiteren Prüfung weitergeleitet (→ OTA-G-120).
R Entwicklungsverantwortlicher A Entwicklungsverantwortlicher C Qualität, Entwickler I (ggf. Regulatorische Behörde / Technischer Dienst)
Output: Prüfergebnis; ggf. qualitätsgeprüftes Update; ggf. Dokumentation zur technischen Lösung (nur bei ernster Gefährdung)
Konditionen: (A) das Software-Update entspricht den Anforderungen des Lastenheftes und besteht die Prüfung durch die *Regulatorische Behörde*. Das Freigabeprozess kann initiiert werden (→ OTA-G-130). **(B)** es entspricht den Anforderungen des Lastenheftes, aber besteht die Prüfung durch die *Regulatorische Behörde* nicht. Der *Entwicklungsverantwortliche* muss das Lastenheft überarbeiten (→ OTA-G-050). **(C)** es entspricht nicht den Anforderungen des Lastenheftes. Der Entwicklungsverantwortliche fordert den Entwickler zur Nachbesserung auf (→ OTA-G-090).
Risiko-E015: Das Update ändert die Fahrzeugfunktionalität über das zugelassene Maß hinaus, sodass die Zulassung erlöschen kann.
UN ECE REQUIREMENT: 7.1.1.7. A process to verify, before a software update is issued, the compatibility of possible software/ hardware configurations for the registered configuration or last known configuration of the target vehicles with the software update; 7.1.1.12.A process whereby the vehicle manufacturer shall be able to make the information according to paragraph 7.1.2.3. and 7.1.2.4. available to relevant Authorities or Technical Services.

OTA-G-120 [nur reg. Beh.]

Input: Dokumentation zur technischen Lösung
Nur bei ernster Gefährdung: Die ***Regulatorische Behörde*** prüft die vorgestellte Lösung auf Konformität.
R Regulatorische Behörde A C Update-Koordinator I

[8] Gemäß der geltenden Auffassung der jeweiligen regulatorischen Behörde: (https://www.kba.de/SharedDocs/FAQ/DE/Marktueberwachung/Produktsicherheit/ernste_Gefaehrdung.html)

Output: Prüfbescheid
UN ECE REQUIREMENT: 8.1.3. Confirmation or extension or refusal of approval, specifying the alterations, shall be communicated by means of a communication form conforming to the model in Annex 2 to this Regulation. […]

OTA-G-130

Input: qualitätsgeprüftes Update, Update-Dokumentation analog UN ECE	
Der ***Entwicklungsverantwortliche*** koordiniert den internen Freigabeprozess.	
R	Entwicklungsverantwortlicher
A	
C	Abteilungen, gemäß interner Richtlinien
I	
Output: freigegebenes Update; Update-Dokumentation analog UN ECE	
UN ECE REQUIREMENT: 7.1.2.5. Documentation for all software updates for that vehicle type describing: 1. The purpose of the update; 2. What systems or functions of the vehicle the update may impact; 3. Which of these are type approved (if any); 4. If applicable, whether the software update affects any of the relevant requirements of those type approved system; 5. Whether the software update affects any system type approval parameter; 6. Whether an approval for the update was sought from an approval body; 7. How the update may be executed and under what conditions; 8. Verification that the software update will be conducted safely and securely. 9. Verification that the software update has undergone adequate verification and validation procedures.	

Erläuterung

Nach erfolgreicher Qualitätsprüfung kann der Freigabe-Prozess initiiert werden. Je nach Organisationsform kann dies länger dauern und ein Hindernis für die notwendige Reaktionsfähigkeit in einem Updateprozesses für kritische Updates sein. Eine Anpassung der Hierarchie für verschiedene Updatetypen kann dies optimieren und einen möglichen Zeitverzug reduzieren. So kann z. B. bei Updates, die nur Security-relevant sind, die Freigabe ausschließlich bei den Security-Experten verortet sein. Eine Freigabe durch den oder die Gesamtfahrzeug-

Verantwortlichen und weitere könnte ggf. entfallen. In der Verantwortung der verkürzten Freigabe-Hierarchie würde es letztlich auch liegen, bei Unsicherheiten diese weiter zu eskalieren.

Hinweis zur Simulation

Der zeitliche Effekt dieser Aktivität ist in der Simulation in der Dauer des gesamten Generierungsprozesses zu berücksichtigen.

OTA-G-141 [neu]

Input: freigegebenes Update, Anforderungskatalog	
Der ***Update-Koordinator*** initiiert die Update-Verteilung durch Hochladen des Updates auf einen Update-Server / ein Backend und benachrichtigt den *Digitalen Dienstleister*. Im Zuge des Uploads zum *Digitalen Dienstleister* prüft der *Update-Koordinator* die initiierten Security-relevanten Mechanismen gegen den Anforderungskatalog auf Erfüllung. Ggf. steuert er weitere, nicht-technische Anforderungen der *Regulatorischen Behörde* ein.	
R	Update-Koordinator
A	Cyber-Security Department
C	
I	Digitaler Dienstleister
Output: freigegebenes und signiertes Update	
`duraForward2CDN`: beschreibt die Zeit, die für die Durchführung dieser Aktivität in Abhängigkeit der internen Prozesse benötigt wird. `duraUpdateMax`: beschreibt die Zeit, die das Update längstens Over-the-Air verteilt wird. Dies kann ggf. durch eine Vorgabe des KBA benötigt werden und bricht die Übertragung via OTA in der Simulation letztlich automatsich ab.	
UN ECE REQUIREMENT: 7.1.3.1. The process they will use to ensure that software updates will be protected to reasonably prevent manipulation before the update process is initiated; 7.1.3.2. The update processes used is protected to reasonably prevent it being compromised, including development of the system update; 7.2.1.1. The authenticity and integrity of software updates shall be protected to reasonably prevent their compromise and reasonably prevent invalid updates.	

Erläuterung

Ziel dieser Aktivität ist es, den sicheren Kanal zwischen Urheber (OEM) und Ziel (Kunde) des Updates abzusichern, so dass angenommen werden kann, dass solange der sichere Kanal besteht, der Inhalt des Paketes bei der Übertragung irrelevant ist.

3.6 Update-Verteilung

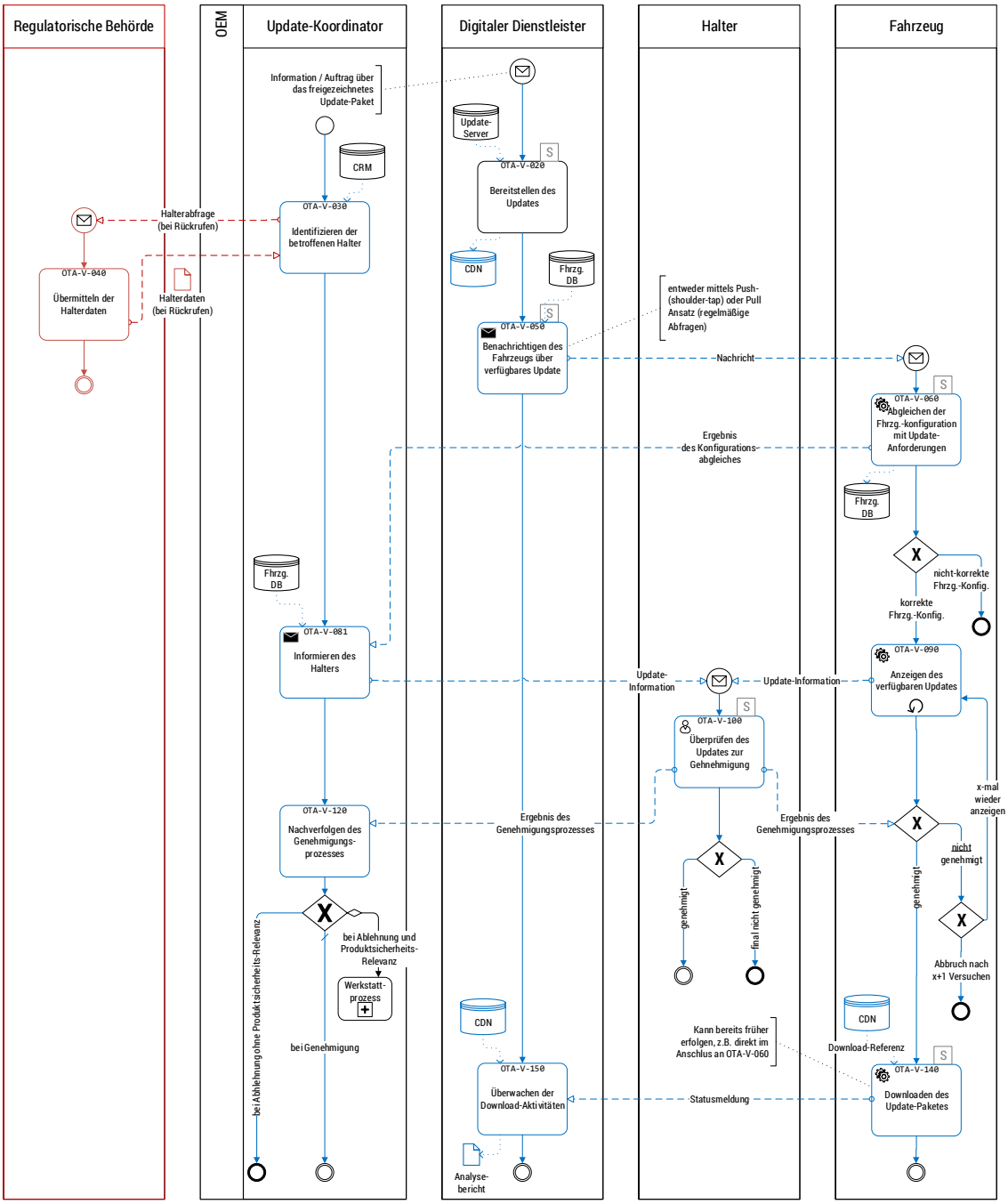


Abbildung 10: Prozessmodellierung "Verteilung"

OTA-V-020

Input: freigegebenes und signiertes Update	
Der *Digitale Dienstleister* stellt das Update auf einer entsprechenden Server-Struktur / einem Content-Delivery-Network (CDN)[9] bereit.	
R	Digitaler Dienstleister
A	
C	
I	Update-Koordinator
Output: im CDN bereitgestelltes Update	
Risiko-E016: Durch den Einsatz eines digitalen Dienstleisters als Intermediär ist das Update Sicherheitsrisiken auf der fremden Infrastruktur ausgesetzt (z. B. Diebstahl von technischem Knowhow oder personenbezogener Daten, fehlende Verfügbarkeit, Verletzung der Integrität, etc.). **Security-Risiko-E017**: Nicht ausreichende Sicherheiten beim digitalen Dienstleister führen zu einem entsprechenden Sicherheitsrisiko: Update-Daten können mitgelesen werden, manipuliertes Update kann aufgespielt werden, Malware kann installiert werden.	
`duraUpload2CDN`: beschreibt die Dauer, die der Digitale Dienstleister zur Mobilisation und Bereitstellung des Updates zum Abruf durch die Fahrzeuge benötigt.	

Erläuterung

Die Infrastruktur des *Digitalen Dienstleister* ist ein mögliches Angriffsziel zur böswilligen Manipulation des Updateprozesses. Hier kann einerseits das Update selbst angegriffen werden oder die Infrastruktur und somit die Kommunikation zwischen dem Backend und der Fahrzeugflotte.

Adressiert werden kann dieses Sicherheitsrisiko durch klare Regelungen über die zu treffenden Sicherheitsmaßnahmen beim *Digitalen Dienstleister*. Dieses kann im Rahmen der Beauftragung erfolgen und durch die Sicherheitszertifizierung von Dienstleistern unterstützt werden.

Weitere technische Maßnahmen, wie zum Beispiel die Partitionierung des Backends, können zusätzlich unterstützen. Solche und andere technischen Maßnahmen bedeuten zwar auf der einen Seite einen höheren Wartungsaufwand für den Digitalen Dienstleister, auf der anderen Seite erhöhen sie aber das Sicherheitsniveau indem sie, beispielsweise durch die Partitionierung, den Zugriff eines Angreifers auf einen Teil der Fahrzeugflotte und die entsprechenden Updates beschränken.

[9] Für die Verteilung des Updates an die Fahrzeuge wird eine neue Serverstruktur benötigt, die eine deutlich höhere Anzahl von Zugriffen bedienen müssen als bei der Verteilung des Updates an die Werkstätten. Gegebenenfalls ist ein Content-Delivery-Network aufzusetzen oder über einen Drittanbieter einzubinden.

Hinweis zur Simulation

Zur Parametrierung der Dauer für diese Aktivität sind interne und externe Faktoren zu berücksichtigen. Entscheidend ist, ob der *Digitale Dienstleister* In-house ist oder ein externer Vertragspartner damit beauftragt wurde. Die Performance dieser Organisation ist entsprechend maßgeblich für die Ausführung der Aktivität, aber auch die Kommunikation zwischen Auftraggeber und Auftragnehmer-Organisationseinheit. Diese kann durch einen effizienten Prozess und das Bereitstellen der notwendigen Informationen verbessert werden. Eine weitgehende Automatisierung kann die Dauer weiter verkürzen.

OTA-V-030

Input: CRM-Tool, Halterdaten (über die *Regulatorische Behörde*)	
Der ***Update-Koordinator*** verknüpft die identifizierten Fahrzeuge mit den entsprechenden Haltern zur korrekten Adressierung. Einerseits kommen die Daten aus dem internen CRM-Tool, andererseits erfolgt ein Abgleich mit der *Regulatorischen Behörde*.	
R	Update-Koordinator
A	
C	Customer Relations
I	Regulatorische Behörde
Output: Halterabfrage / FIN-Liste identifizierter Fahrzeuge	
Konditionen: **(A)** Wenn Behörden-relevant, dann sind die Halterdaten abzufragen, um ggf. die Halter auch auf Kanälen außerhalb des Fahrzeuges zu erreichen und über das Update zu informieren. **(B)** Diese Aktivität kann auch schon früher stattfinden, parallel zur Updateentwicklung (z. B. analog mit OTA-G-060)	

Erläuterung

Zu klären ist, welcher Kanal zur Information des Kunden – auch bei Rückrufen – hinreichend ist. Mit der Kommunikation im Fahrzeug ist nicht immer der Halter zu erreichen und damit eventuell auch keine rechtsgültige Genehmigung des Updates. Darüber hinaus ist die Verknüpfung der Halterdaten notwendig, um zu überprüfen, unter welcher regulatorischen Aufsicht sich das Fahrzeug aktuell befindet (z. B. bei Wiederverkauf des Fahrzeuges ins Ausland).

Hinweis zur Simulation

Diese Aktivität wurde nicht simuliert, weil sie eine parallele Aktivität zu kritischen Aktivitäten ist. Ihr Einfluss auf die Gesamtdurchlaufzeit des Prozesses ist nachrangig. Auch bei sicherheitskritischen Updates ist dies kein Hindernis, da die Fahrzeuge auch ohne diese Aktivität adressiert werden können und das Update erhalten können.

OTA-V-040 [optional] [nur regel. Beh.]

Input: Halterabfrage
Die ***Regulatorische Behörde*** übermittelt bei Bedarf die Halterdaten zu den betroffenen Fahrzeugen.
R Regulatorische Behörde A C I Update-Koordinator
Output: Halterdaten

Erläuterung

Diese Aktivität ist mit aufgeführt, da sie letztlich ein zeitlicher Baustein im Gesamtprozess sein könnte – abhängig vom regulatorischen Rahmen.

OTA-V-050

Input: Fahrzeug-Datenbank
Der ***Digitale Dienstleister*** benachrichtigt die Fahrzeuge über das verfügbare Software-Update. Entweder erfolgt dies direkt über einen ‚Shoulder-Tap' oder indirekt, indem das *Fahrzeug* in definierten Zeitintervallen überprüft, ob Updates verfügbar sind (‚Pull-Methode').
R Digitale Dienstleister A Update-Koordinator C Update-Koordinator I Fahrzeug
Output: Nachricht
Security-Risiko-E018: Die Nachricht kann so manipuliert werden, dass das Fahrzeug angewiesen wird, ein Roll-Back auf eine vorherige, möglicherweise fehlerhafte Version durchzuführen. **Risiko-E019**: Die Nachricht erreicht das Fahrzeug nicht, so dass das Fahrzeug in einem unsicheren Zustand verbleibt. Ggf. ist ein Verbindungsaufbau zum Fahrzeug nicht möglich (technischer Defekt, Privacy Modus des Halters, o.ä.). **Chance-E020**: Durch hohe verfügbare Kapazitäten können mehr Fahrzeuge gleichzeitig adressiert und aktualisiert werden, als über die existierenden Werkstatt- und Vertriebsnetzwerke.

`methodCallVehicle`: differenziert zwischen der Shoulder-Tap und Pull-Variante

`duraToCallVehicle`: die Dauer, um eine Nachricht sowohl auf dem Server (Shoulder-Tap Methode) als auch im Fahrzeug (Pull Methode) zu initiieren

`pSuccessCallVehicle`: die Wahrscheinlichkeit, dass ein Fahrzeug erreicht wird / die Verbindung aufgebaut werden kann

`maxCallsAtATime`: die maximale Anzahl von Nachrichten an ein Fahrzeug, die zu einem Zeitpunkt aufeinanderfolgend abgeschickt werden, sollte keine direkte Antwort erfolgen

`duraNextCalls`: die Dauer zwischen zwei Benachrichtigungszeitpunkten

`maxCallAttempts`: die maximale Anzahl von Nachrichten an ein Fahrzeug bevor der OTA-Prozess abgebrochen wird und an den Werkstatt-Prozess übergeben wird

`nDaysToAskForUpdate`: Anzahl der Tage, in deren Rhythmus das Fahrzeug nach Updates am Server anfragt (nur Pull Methode)

Erläuterung

Dieser Prozessschritt kann in zwei Varianten durchgeführt werden. Entweder die Fahrzeuge werden direkt mittels eines sog. ‚Shoulder-Tap' benachrichtigt, indem an jedes Fahrzeug eine Nachricht gesendet wird, die dem Fahrzeug mitteilt, dass ein Update für sie zur Verfügung steht. Die andere Variante entspricht einer ‚Pull'-Konfiguration, bei der die Fahrzeuge selbstständig in regelmäßigen Abständen bei vorhandener Datenverbindung das Backend nach verfügbaren Updates abfragen. Die Methode muss vor Durchführung des Prozesses festgelegt werden (`methodCallVehicle`). Vorteilhaft an der Shoulder-Tap Methode ist die Geschwindigkeit, mit der prinzipiell alle Fahrzeuge erreicht werden könnten, nachteilig ist jedoch, dass hierfür eine höhere Serverkapazität bereitstehen muss. Zusätzlich ist die Shoulder-Tap Methode auch schwieriger zu steuern ist, da die Zugriffsraten deutlich volatiler sind als bei der Pull Methode.

Bei der Pull Methode greifen die Fahrzeuge über einen längeren Zeitraum auf den Server zurück, wodurch eine geglättete Zugriffskurve entsteht. Entsprechend kann der Bedarf kostengünstiger optimiert werden und sogar die Zugriffszeiten verkürzt werden, da Serverüberlastungen oder -engpässe einfacher vermieden werden können.

Durch die verzögerte Austeilung via Pull-Methode kann auch bei auftretenden Fehlern ggf. nur eine geringere Menge an Fahrzeugen betroffen sein, bevor der Rückruf des Updates erfolgt. Eine solche Beta-Phase, dass zunächst nur ein Teil der Flotte aktualisiert wird, kann jedoch auch geplant werden und ist dann generell unabhängig von beiden Methoden, mittels Shoulder-Tap aber effektiver, da der zeitliche Verzug bis zur Rückmeldung des letzten Fahrzeuges geringer sein könnte.

Die Wahrscheinlichkeit, dass ein Fahrzeug erreicht wird / die Verbindung aufgebaut werden kann (`pSuccessCallVehicle`), ist u. a. abhängig, vom Fahrzeugtyp und Nutzertyp. Ein Smart-EV hat primär als Stadtauto ein anderes Profil, als ein Pendler-Fahrzeug, das häufiger über Land fährt. Lange Überlandfahrten beeinflussen zum Beispiel die Erreichbarkeit des Fahrzeugs durch eine variierende Netzabdeckung im Mobilfunknetz. Die Abdeckung durch High-Speed Mobilfunk ist aufgrund der höheren Nachfrage in der Stadt dagegen deutlich höher. Andererseits steht hier

deutlich seltener private Ladeinfrastruktur zur Verfügung, die ggf. neben Strom auch häufig WiFi-Anbindungen anbieten und so wiederrum die Erreichbarkeit der Fahrzeuge erhöhen.

OTA-V-060

Input: Nachricht über verfügbares Update
Das ***Fahrzeug*** gleicht die Fahrzeugkonfiguration mit den Update-Anforderungen ab (verbaute Hardware, installierte Software, etc.) und prüft ob sich inzwischen durch einen Werkstattaufenthalt (z. B. nach einem Unfall) die Fahrzeugkonfiguration verändert hat oder sich der Softwarestand durch vorausgegangene Updates verändert hat. Der Prozess findet automatisiert statt.
R Fahrzeug A Update-Koordinator C Digitaler Dienstleister I Update-Koordinator
Output: Ergebnis des Konfigurationsabgleichs
Risiko-E021: Die Rahmenbedingungen, unter denen Daten aus dem Fahrzeug abgerufen werden dürfen, sind zu klären. **Security-Risiko-E022**: Die Datenschnittstelle und / oder andere Fahrzeugelektronik können über die Datenschnittstelle manipuliert oder missbraucht werden. Dies kann z. B durch eine Denial-of-Service (DoS) Attacke geschehen, so dass die Nachricht nicht empfangen oder verarbeitet werden kann, da die Connectivity-Unit mit der Verarbeitung zu vieler Anfragen überlastet ist. **Chance-E023**: Die häufige Kommunikation mit den Fahrzeugen stellt eine stets aktuelle Datenbasis sicher.
`duraCheckConfig`: die Dauer, um fahrzeugseitig die Fahrzeugkonfiguration zu überprüfen und dem Server zu antworten. `pSuccessConfig`: die Wahrscheinlichkeit dafür, dass Fahrzeug-Konfiguration korrekt ist.

Erläuterung

Diese Aktivität findet automatisiert statt und ihre Robustheit ist somit von der Qualität der technischen Implementation der Funktion im Fahrzeug abhängig. Dies beeinflusst auch die Simulation und die Güte des Prozessergebnisses.

OTA-V-081 [neu]

Input: Ergebnis des Konfigurationsabgleichs, aktualisierte Fahrzeug-Datenbank
Der ***Update-Koordinator*** innformiert den *Halter* über das Update.
R Update-Koordinator A C I Halter
Output: Update-Information
Chancen-E024: Die Möglichkeit der einfacheren und schnellen Veränderung von Software ermöglicht neue Geschäftsmodelle (z.B. Function on Demand). **Chance-E025:** Durch die Durchführung des Software-Updates ohne die Partizipation einer Werkstatt sind Kostenersparnisse und Zeitersparnisse möglich. **Risiko-E028:** Eine unzureichende Information des Fahrzeugeigentümers über das Software-Update könnte einen unerlaubten Eingriff in die Eigentumsrechte des Eigentümers darstellen. **Security-Risiko-E030:** Die Kommunikation des Updates zum aktuellen Halter des Fahrzeuges sind unter besonderer Berücksichtigung von Privacy-Anforderungen durchzuführen. Bei nachträglichen Funktionsfreischaltungen könnten z.B. Zahlungsdaten kompromittiert werden. **Risiko-E031**: Updates, die bzgl. Zeit oder Sicherheit kritischer sind, werden nicht atomar verpackt und werden ggf. nicht berücksichtigt.
UN ECE REQUIREMENT: 7.1.1.11 A process whereby the vehicle user is able to be informed about updates. 7.2.2.2 The vehicle manufacturer shall demonstrate that the vehicle user is able to be informed about an update before the update is executed. The information provided may contain: • The purpose of the update. This could include the criticality of the update and if the update is for recall, safety and/or security purposes; • Any changes implemented by the update on vehicle functions; • The expected time to complete execution of the update; • Any vehicle functionalities which may not be available during the execution of the update; • Any instructions that may help the vehicle user safely execute the update; • In case of groups of updates with a similar content one information may cover a group.

Erläuterung

Die Informationen über das Update kann auf verschiedenen Wegen stattfinden: per Postwurfsendung, per E-Mail, On-Screen im Fahrzeug, Smartphone-Applikation, etc. Die Updateinhalte sollten dem Kunden eindeutig und in verständlicher Sprache geschildert werden, um ihm über die Veränderungen am Fahrzeug und über die Dauer zu informieren, während der das Fahrzeug dem Kunden aufgrund des Updates nicht zur Verfügung steht.

Bei der Information für den Kunden über ein Rückruf-Update ist eindeutig hervorzuheben, dass es sich um eine Rückrufmaßnahme handelt, die bis zu einem gewissen Zeitpunkt durchgeführt

werden muss. Das Update darf durch den Kunden nur solange zurückgehalten werden, wie es der entsprechende Zeitraum definiert, anschließend müsste das Fahrzeug in der Werkstatt vorgeführt werden oder das Fahrzeug verliert ggf. seine Zulassung.

OTA-V-090

Input: Update-Information, korrekte Fahrzeugkonfiguration
Das ***Fahrzeug*** zeigt das verfügbare Update über das In-Vehicle-Infotainment an und fragt ggf. vom *Halter* die Genehmigung für das Update an.
R Fahrzeug A Update-Koordinator C I Halter
Output: Update-Information (z. B. Informationsscreen mit Möglichkeit zur Bestätigung / Verzögerung / Ablehnung des Updates)
Risiko-E032: Fehlerhafte Informationen über das Update (z.B. Dauer der Installation) führen zu einem negativen Erlebnis der Kunden. **Security-Risiko-E033:** Die Updatefunktion kann blockiert werden, z.B. durch Denial-of-Service Attacken (Überlasten des Datennetzes oder Überlasten der Empfangseinheit), so dass das Fahrzeug in einem unsicheren Zustand verbleibt.

Erläuterung

Dem Kunden können bei einem Update die Möglichkeiten angeboten werden, das Update durchzuführen, später erneut daran erinnert zu werden oder das Update abzulehnen. Eine entsprechende ausführliche und vor allem auch hinreichende Information des Kunden ist wichtig. Ziel ist es – unter andere –, dass der Kunde sicherheitskritische Updates als solche wahrnimmt und entsprechend darauf reagiert, indem er der Installation zeitnah zustimmt. Dies kann erreicht werden, indem der Kunde direkt über die entscheidenden Parameter informiert wird: Dauer, die das Update benötigt; mögliche Kosten, die für die Übertragung anfallen; Aktivitäten, die der Fahrer durchführen muss; etc. Eine einfache und direkte Sprache sowie entsprechende Menüführung unterstützen dies zusätzlich.

Grundsätzlich ist zu berücksichtigen, dass bei der Information im Fahrzeug, die Update-Information auch Personen erreichen können, die nicht für die Updategenehmigung mandatiert sind.

OTA-V-100

Input: Updateinformation	
Der ***Halter*** überprüft, die ihm zur Verfügung gestellten Informationen und entscheidet, ob das Update zur Installation genehmigt wird.	
R	Halter
A	Halter
C	
I	Digitaler Dienstleister, Update-Koordinator
Output: Ergebnis des Genehmigungsprozesses (Zustimmung, Rückstellung oder finale Ablehnung)	
Entweder er genehmigt das Update **oder** er genehmigt das Update nicht. Im negativen Falle kann das Update bis zu x-mal wiederholt angezeigt werden. Sollte das Update auch weiterhin nicht genehmigt werden, wird der Updateprozess Over-The-Air abgebrochen und (bei Bedarf / Relevanz) im konventionellen Werkstattprozess weiter nachverfolgt.	
Chance-E034: OTA-Updates reduzieren den Aufwand für den Fahrzeugkunden bei Software-Updates. **Risiko-E035:** Unautorisierte Personen genehmigen ein Update, so dass vom Eigentümer ungewollte Software / Updates im Fahrzeug installiert werden. **Risiko-E036:** Bei einer Ablehnung des Updates durch den Halter kann das Fahrzeug weiter in einem vielleicht sicherheitskritischen Zustand verbleiben. **Risiko-E037:** Bei einer Installation ohne vorliegende Einverständniserklärung kann ein unerlaubter Eingriff in die Eigentumsrechte des Eigentümers vorliegen.	
`duraRfC`[10]: Dauer, die der Genehmigungsprozess in Anspruch nimmt. Dies wird beeinflusst durch die technische Umsetzung und Kommunikation sowie durch die Informationsverarbeitung durch den Halter. `duraNextRfC`: Dauer, bis nach einem Zurückstellen des Updates, die Verfügbarkeit des Updates wieder im Fahrzeug angezeigt wird. `maxDuraRfC`: maximal Dauer, die ein Update angezeigt wird, bevor zum Beispiel der Werkstattprozess initiiert wird. `maxRfC`: maximale Anzahl, die ein Update zurückgestellt werden kann. (Kann auch aus `duraNextRfC` und `maxDuraRfC` berechnet werden oder durch diese vorgegeben werden.) `pSuccessRfC`: Wahrscheinlichkeit dafür, dass ein Kunde das Update genehmigt.	

Erläuterung

Eine Implementation der Genehmigung des Updates von außerhalb des Fahrzeuges ist zu empfehlen. Über den Fahrzeugschlüssel, eine Smartphone-App o. ä. kann der Halter über Updates informiert werden und er kann diese initiieren während er nicht am Fahrzeug ist. Auf diese Weise hat der Halter mehr Möglichkeiten den Updatevorgang in seinen Tagesablauf zu integrieren und nicht mehr nur während der Fortbewegung daran erinnert zu werden. Die Wahrscheinlichkeit, dass ein Update zeitnah genehmigt wird, steigt so.

[10] RfC: Request for Conformation (projektspezifische Bezeichnung)

Zusätzlich kann anhand einer Analyse des Fahrprofils dem Eigentümer ein geeigneter Zeitpunkt vorgeschlagen werden, zu dem das Fahrzeug nicht benutzt wird und ein Update ohne Einschränkung durchgeführt werden kann.

Eine saubere Dokumentation und Informationsaufbereitung gegenüber dem Kunden erhöht die Wahrscheinlichkeit, dass der Kunde das Update installiert, und verkürzt die Verarbeitungsdauer des Kunden bis zur Genehmigung. Selteneres Zurückstellen des Updates wird so wahrscheinlicher. Hier ist außerdem ein Gleichgewicht zu finden, zwischen der *notwendigen* Häufigkeit der Benachrichtigung des Kunden, um ein Update schnellstmöglich genehmigen bzw. installieren zu lassen, und der *möglichen* Häufigkeit, dass sich der Kunde nicht von der Update-Informationen gestört fühlt.

Ein weiterer Faktor, der in dieser Aktivität zu berücksichtigen ist, sind die jeweiligen Nutzergruppen der Fahrzeuge, die das Update adressiert. Bei Gruppen, die weniger Technik-affin sind, müssen die Informationen anders in Sprache und Aufbereitung kommuniziert werden. Entsprechend geschulte Service-Mitarbeiter, könnten explizit als Ansprechpartner zur Verfügung stehen, um bei Rückfragen, die Genehmigung zu unterstützen. Zusätzlich sind Technik-affine Kunde eventuell aufgeschlossen gegenüber neuen Software-Versionen und ihren neuen Funktionen (analog sog. Early-Adapters).

Aus rechtlicher Sicht ist zu beachten, dass die Person, die angesprochen wird, auch ein Mandat hat, das Update am Fahrzeug zu genehmigen. Zum Beispiel ist dies bei Flottenfahrzeugen in der Regel nicht der Fahrer.

OTA-V-120

Input: Ergebnis des Genehmigungsprozesses
Der ***Update-Koordinator*** verfolgt den Genehmigungsprozess des *Halters* nach.
R (Update-Koordinator) A Update-Koordinator C I
Output: ggf. Initiierung des Werkstattprozesses bei Ablehnung
Anm.: Dies kann eine automatisierte Aktivität sein. Die Überwachung bzw. die Verantwortung, dass alle Anforderungen erfüllt werden – z. B. die der *Regulatorischen Behörde* – liegt beim *Update-Koordinator*.
Konditionen: **(A)** bei erteilter Genehmigung wird der Digitale Dienstleister entsprechend benachrichtigt. **(B)** bei nicht erfolgter Genehmigung wird entweder der OTA-Prozess beendet (bei einem Update ohne Produktsicherheits-Relevanz) oder **(C)** beendet und an den Werkstattprozess übergeben (bei einem Update mit Produktsicherheits-Relevanz).

OTA-V-140

Input: Ergebnis des Genehmigungsprozesses
Das ***Fahrzeug*** lädt das Update-Paket über eine Luftschnittstelle in das System. Nach Erhalt einer positiven Rückmeldung aus dem Genehmigungsprozess wird ein Skript ausgeführt und eine entsprechende Downloadreferenz abgefragt, über die das Updates geladen wird. Alternativ kann das Updatepaket auch über ein Third-Party-Device geladen (und zwischengespeichert) werden, welches per Kabel mit dem Fahrzeug verbunden wird und so das Update aufspielt („Tethering“).
R Fahrzeug, digitaler Dienstleister A Update-Koordinator C (Halter) I Halter, Update-Koordinator, (regulatorische Behörde)
Output: lokal im *Fahrzeug* gespeichertes Update
Wenn zeitkritisch: a) dann sollte das Updates vorgezogen werden und schon vorab der Genehmigung des Fahrers heruntergeladen werden. b) dann sollte das Update über Mobilfunk verteilt werden und nicht ausschließlich über WiFi. Wenn Safety-relevant: c) dann sollten zusätzliche Security-Maßnahmen ergriffen werden. (z. B. die Übertragung zum Fahrzeug wird mittels eines eigenen Funkkanals des Netzanbieters realisiert.)
Risiko-E037: Das Fahrzeug kann den Download nicht initiieren und das Update kann nicht geladen werden, z.B. aufgrund einer Überlastung der Serverinfrastruktur oder aufgrund einer fehlerhaften Connectivity-Unit. **Risiko-E038**: Das Update wird unvollständig oder fehlerhaft übertragen. **Security-Risiko-E039**: Die Datenschnittstelle und/oder andere Fahrzeugelektronik werden manipuliert oder missbraucht **Security-Risiko-E039b**: Bei Tethering sind die entsprechenden Devices nicht im Einflussbereich / Absicherungsbereich des Herstellers, so dass über kompromittierte Geräte Schäden verursacht werden könnten.
`nDownloadSlots`: die Anzahl der zur Verfügung stehenden Slots für den Download. Entspricht der Simulation der Bandbreite. Ein Slot kann ein Fahrzeug bedienen. `pSuccessDownload`: Die Wahrscheinlichkeit für den Erfolg eines Updates kann (A) durch eine bessere Kommunikation oder (B) durch Verwendung von Mobilfunk-Netzen (ggf. zu Lasten der eigenen Kosten) verbessert werden. Bzw. durch eine Aktivierung des Downloads durch eine Smartphone-Application. `duraDownloadIdle`: Diese Zeit kann durch den Einbau besserer Hardware verkürzt werden.
UN ECE REQUIREMENT: 7.2.1.1. The authenticity and integrity of software updates shall be protected to reasonably prevent their compromise and reasonably prevent invalid updates.

Erläuterung

Zur Steigerung der Kundenzufriedenheit kann das Update bereits direkt nach der Freischaltung heruntergeladen werden, d.h. noch vor der eigentlichen Autorisierung des Updates durch den Halter. Eine Installation kann somit verzögerungsfrei direkt nach der Autorisierung erfolgen. Dies ist jedoch ggf. mit höheren Kosten verbunden, z.B. bei einer dann nicht folgenden Autorisierung des Updates oder wenn der OEM die Kosten der Mobilfunk-Übertragung selbst trägt. Anstelle des Kunden, der eine entsprechende Option beim OEM erworben hat oder durch einen eigenen mobilen Hotspot die Mobilfunkverbindung bereitstellt.

Ein Update kann auch in mehreren ‚Wellen' an die Fahrzeuge verteilt werden, um die Zugriffe auf die Serverinfrastruktur zeitlich zu verteilen. Bei nicht zeitkritischen Updates kann so auch zunächst die Funktionsweise im Feld überprüft werden.

Hinweis zur Simulation

Für die zu aktualisierenden Fahrzeuge stehen serverseitig eine Anzahl an Slots zur Verfügung, von denen parallel das Update abgerufen werden kann (`nDownloadSlots`). Die Entscheidung, dass ein Update erfolgreich nach der Belegung eines freien Slots heruntergeladen wird, ist abhängig von einem Wahrscheinlichkeitsfaktor (`pSuccessDownload`). Dieser ist wiederum abhängig von der Kundengruppe und der Technik (WiFi hat eine deutlich geringe Abdeckung und eine langsamere Verbindungsgeschwindigkeit, so dass z. B. durch die Weiterfahrt des Kunden die Verbindung abbricht oder erst gar kein Netz zur Verfügung steht). Ist das Update erfolgreich wird dies entsprechend in der Datenbank vermerkt. Je mehr Slots für den Download zur Verfügung stehen, desto schneller kann das Update geladen werden – allerdings bei gleichzeitig steigenden Kosten. Die Option der Eingrenzung der Slots dient auch in der Simulation dazu, die Bandbreite bei mehreren parallelen Updates einzuteilen und entsprechend der Priorisierung der Updates zu optimieren.

Schlägt der Download fehl wird der Download zu diesem Zeitpunkt mehrfach versucht (`maxDownloadAttempts`). Wird diese Anzahl überschritten, wird ein weiteres Herunterladen zunächst nicht versucht. Die Fahrzeugreferenz wird auf einem Stack nach der First-In-First-Out Logik zwischengespeichert. Nach einer festgelegten Zeitspanne erfolgt der Zugriff vom Stack auf einen Slot für den nächsten Download-Versuch (`duraNextDownload`). Nach einer maximalen Anzahl an Download-Versuchen wird die Fahrzeugreferenz aus dem OTA-Prozess entfernt (`maxDownloadAttempts`). Die Aktualisierung muss entweder über eine Werkstatt oder manuell vorgenommen werden.

Unabhängig vom Erfolg setzt sich die Dauer eines Download-Versuchs aus der Übertragungsdauer in Abhängigkeit von der Übertragungstechnik (3G, 4G, 5G oder WiFi) und aus der fahrzeugseitigen und serverseitigen Performance-Zeit zusammen (`duraDownloadIdle`). Diese Zeit wird auf beiden Seiten des Übertragungskanals benötigt, um das Update zu verarbeiten und zusätzlich im Fahrzeug zu validieren.

In der Simulation ist ebenfalls die Option implementiert, dass Kunden das Update via Tethering zum Fahrzeug übertragen: entweder indem sie ihr Mobiltelefon als mobilen Hotspot zur Datenübertragung verwenden oder indem sie das Update über einen PC oder Laptop auf einen USB-Stick laden und dann diesen ans Auto anschließen. Durch das Öffnen einer solchen Schnittstelle entstehen weitere Risiken, da die Tethering-Geräte der Kunden eventuell nicht ausreichend gesichert wurden und Schadsoftware enthalten, die das Fahrzeug nach dem Anschließen des Gerätes ebenfalls bedrohen könnten. Die Absicherung solcher nicht-vertrauenswürdigen Geräte ist bei einem solchen Angebot besonders zu berücksichtigen.

OTA-V-150 [neu]

Input: Statusmeldung	
Das ***Digitale Dienstleister*** überwacht die Download-Aktivitäten bzgl. möglicher Engpässe, Fehlermeldungen oder sonstigem Feedbacks	
R	Digitaler Dienstleister
A	Digitaler Dienstleister
C	Halter, Fahrzeug
I	Update-Koordinator, (regulatorische Behörde)
Output: Analysebericht	

Erläuterung

Eine nachhaltig implementierte Monitoring-Aktivität ist wichtig, um rechtzeitig auf Unregelmäßigkeiten im Updateprozess reagieren zu können. Im besten Fall werden bei einem schwerwiegenden Fehler so deutlich weniger Kunden davon betroffen sein.

3.7 Update-Installation

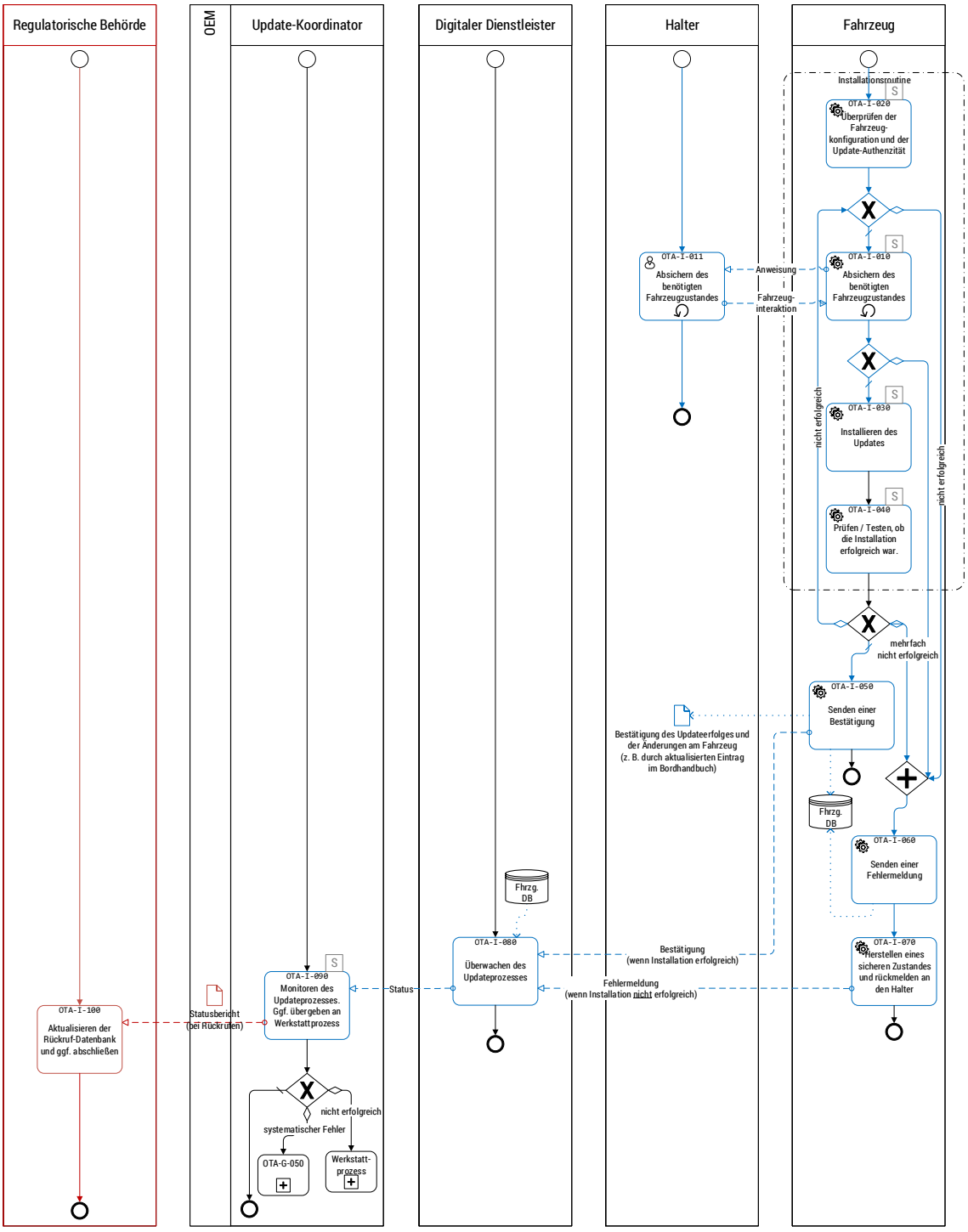


Abbildung 11: Prozessmodellierung "Installation"

OTA-I-020

Input: lokal im *Fahrzeug* gespeichertes Update
Das ***Fahrzeug*** überprüft, ob es die Anforderungen und die Fahrzeugkonfiguration für das Update erfüllt (Speicherplatz, notwendige Software-Bibliotheken etc.), und es überprüft die Authentizität des Updates.
R Fahrzeug A Entwicklungsverantwortlicher C I
Output: Bestätigung der korrekten Fahrzeugkonfiguration; ggf. Forderung nach weiteren Softwarepaketen
Entweder werden die Anforderungen erfüllt (→ OTA-I-030) **oder** die Überprüfung schlägt fehl. Bei negativem Ausgang wird eine Fehlermeldung an das Backend gesendet (→ OTA-I-060) **oder** es fehlen Voraussetzungen, die aber durch das Nachladen von Software erfüllt werden können. Dies bedeutet nur eine Verzögerung und keinen Abbruch des Prozesses.
Risiko-E040: Ein Software-Update wird aufgespielt, das nicht mit dem Fahrzeug kompatibel ist, sodass es zu Funktionalitätsverlusten kommt. **Risiko-E041:** Das Update kann aufgrund fehlender Betriebsressourcen nicht aufgespielt werden oder es bricht während der Installation ab, sodass das Fahrzeug in einem nicht gewünschten Zustand verbleibt. **Security-Risiko-E042:** Es wird unautorisierte, manipulierte oder gefälschte Software zur Installation freigegeben.
`duraCheckConfig`: Dauer, die fahrzeugseitig benötigt wird, um die Fahrzeugkonfiguration zu überprüfen und dem Server zu antworten. (analog OTA-V-060) `pSuccessConfig`: Wahrscheinlichkeit dafür, dass Fahrzeug-Konfiguration korrekt ist. (analog OTA-V-060)
UN ECE REQUIREMENT: 7.2.1.1. The authenticity and integrity of software updates shall be protected to reasonably prevent their compromise and reasonably prevent invalid updates.

Erläuterung

Ggf. fehlen dem Fahrzeug für das Update notwendige Standard-Softwarebibliotheken, die entsprechend vor der Durchführung des Updates zunächst geladen und installiert werden müssen. Dies würde wiederum einem eigenen Updateprozess entsprechen und ist deswegen weiter nicht berücksichtigt.

Wenn das Fahrzeug eine nicht-passende Konfiguration aufweist[11], z. B. andere ECUs als für das Update angedacht, wird das Update als fehlgeschlagen markiert und entsprechend ans Backend gemeldet.

[11] Die Differenz zwischen dem aktuellen Fahrzeugzustand und einer vorhergehenden Konfigurationsabfrage kann zum Beispiel durch einen Unfall begründet sein.

OTA-I-010

Input: lokal im *Fahrzeug* gespeichertes Update und Bestätigung der korrekten Fahrzeugkonfiguration
Das ***Fahrzeug*** führt eine Routine durch, um den benötigten Fahrzeugzustand abzusichern (z. B. Batterie-SoC, Zündung, Fensterposition, Gang, etc.). Bedarf ein Kriterium der Bedienung durch einen Benutzer oder wird ein Kriterium nicht erfüllt, wird dies dem Fahrzeuginsassen kommuniziert und nach Bestätigung erneut geprüft. Ist der Kunde nicht erreichbar, muss das Update zurückgestellt werden.
R Fahrzeug A Entwicklungsverantwortlicher C I Halter
Output: Bestätigung des korrekten Zustandes, Anweisung an den Kunden oder Fehlermeldung
Entweder der benötigte Fahrzeugzustand kann erfolgreich hergestellt werden (→ OTA-I-020) **oder** es bedarf der Unterstützung durch einen Benutzer (OTA-I-011) **oder** die Routine ist nicht erfolgreich. Bei negativem Ausgang wird eine Fehlermeldung an das Backend gesandt (→ OTA-I-060).
Wenn sicherheitskritisch: a) dann höchste Anforderungen an den Fahrzeugzustand. Ggf. ist dies für Security-relevante Updates nicht erforderlich, wenn keine Safety-relevanten Bereiche angesprochen werden. Wenn zeitkritisch: b) dann muss in einem hohen Intervall der Fahrer häufig an das Herstellen des notwendigen Zustandes des Fahrzeuges erinnert werden.
`maxChecksPerSession`: maximale Anzahl von Versuchen einer Zustandsprüfung pro Zeitpunkt `maxSessionsPerDay`: maximale Anzahl Prüfungszeitpunkte pro Tag `maxChecksPerVehicle`: maximale Anzahl von Zustandsprüfungen bis zum Abbruch des Prozesses `pSuccessVehicleState`: Wahrscheinlichkeit dafür, dass der Fahrzeugzustand korrekt ist `duraCheckState`: Dauer zum Prüfen des Fahrzeugzustandes
UN ECE REQUIREMENT: 7.1.4.1. The vehicle manufacturer shall demonstrate the processes and procedures they will use to assess that over the air updates will not impact safety if conducted during driving. 7.2.2.1.2The vehicle manufacturer shall ensure that software updates can only be executed when the vehicle has enough power to complete the update process (including that needed for a possible recovery to the previous version or for the vehicle to be placed into a safe state). 7.2.2.3. In the situation where the execution of an update whilst driving may not be safe, the vehicle manufacturer shall demonstrate how they will: ▪ Ensure the vehicle cannot be driven during the execution of the update; ▪ Ensure that the driver is not able to use any functionality of the vehicle that would affect the safety of the vehicle or the successful execution of the update

Hinweis zur Simulation

Die Anzahl der Versuche zum Testen des Fahrzeugzustandes ist stark abhängig vom Nutzerprofil. In der Simulation kann dies über die Wahrscheinlichkeit adressiert werden. Außerdem wird die Wahrscheinlichkeit von der Robustheit der Aktivität selbst und den Anforderungen durch das Update bestimmt. Bei einer Installationsroutine, die z B. durch eine App ausgelöst wurde, ist der Nutzer zum gewählten Installationszeitpunkt nicht zwangsläufig am Fahrzeug, so dass ein fehlerhafter Zustandsparameter

OTA-I-011 [neu]

Input: Anweisung	
Der ***Halter*** führt die vom Fahrzeug geforderten Maßnahmen zum Absichern des Fahrzeugzustandes durch.	
R	Halter
A	Update-Koordinator
C	
I	Fahrzeug
Output: Fahrzeuginteraktion	

Erläuterung

Diese Aktivität durch den Kunden ist gesondert aufgeführt, da sie ein Unsicherheitsfaktor darstellt, der durch geeignete Kommunikation mit dem Kunden reduziert werden kann. Hierzu zählen die Einfachheit der Sprache und der Anweisungen im In-Vehicle-Infotainment.

OTA-I-030

Input: lokal im Fahrzeug gespeichertes Update, Bestätigung der korrekten Fahrzeugkonfiguration und des korrekten Fahrzeugzustandes	
Das ***Fahrzeug*** entschlüsselt, entpackt und installiert selbstständig das Update.	
R	Fahrzeug
A	Entwicklungsverantwortlicher
C	
I	Halter
Output: Signal, dass die Installation durchgeführt wurde.	
Chance-E043: Die Installation findet selbstständig ohne notwendiges Personal und ohne entsprechende Kosten statt.	

Risiko-E044: Die Installation bricht ab (z.B. durch eine Veränderung des Fahrzeugzustandes) und das Fahrzeug verbleibt in einem ungewünschten Systemzustand. **Security-Risiko-E045:** Nach der Entschlüsselung des Updates auf dem Gateway, kann das Update und damit Know-How – z.B. auf dem CAN-BUS mitgelesen werden.
`powerECU`: Leistungsfähigkeit der eingesetzten Rechnereinheit `pSuccessInstall`: Wahrscheinlichkeit dafür, dass das Update erfolgreich installiert wird `duraNextInstallation`: Dauer bis die nächste Installation (nach einer fehlgeschlagenen) versucht wird. `maxInstallsPerVehicle`: maximale Anzahl an Installationsversuchen

Erläuterung

Die Leistungsfähigkeit der Ziel-ECU sowie die Paketgröße beeinflussen hier maßgeblich die Dauer der Aktivität. Vor allem das Flashen von weniger leistungsfähigen Steuergeräten kann entsprechend länger dauern. Eine Maßnahme wäre zum Beispiel die Entschlüsselung durch eine leistungsfähigere CPU – z. B. die des Infotainments – durchführen zu lassen. Nachteilig wäre allerdings, dass dann das Updatepaket unverschlüsselt zur Ziel-ECU gesandt wird und so im Bordnetzwerk mitgelesen werden kann.

OTA-I-040

Input: Signal, dass die Installation durchgeführt wurde.	
Das ***Fahrzeug*** prüft / testet, ob die Installation erfolgreich war. Entsprechende Testmechanismen sind zu implementieren, die die Installationsroutine und ggf. anschließend die veränderte Funktionalität überprüfen.	
R	Fahrzeug
A	Entwicklungsverantwortlicher
C	
I	
Output: Testergebnis der Installation	
Entweder die Installation war erfolgreich **oder** die Installation schlug fehl. Bei einem Fehler kann zunächst versucht werden, die Installationsroutine zu wiederholen (→ OTA-I-010). Schlägt die Installation mehrfach fehl, wird eine Fehlermeldung erstellt (→ OTA-I-060).	
Wenn Safety-relevant: a) ist ein Testverfahren der Safety-relevanten Funktionen zu empfehlen, das prüft, ob die durch die Installation betroffenen Funktionen auch wie vorgesehen funktionieren.	
Risiko-E046: Nach der Installation oder einem Abbruch der Installation wird ein Fehler festgestellt, sodass das Fahrzeug nicht mehr fahrbereit ist. Eine Rückführung in einen fahrbereiten Zustand ist vorzunehmen.	

Risiko-E046b: Nach der Installation wird deren Abbruch oder ein Fehler nicht festgestellt, so dass es zu Schäden durch eine Fehlfunktion kommen kann.
`duraCheckInstall`: Dauer, die zum Durchführen des Checks benötigt wird.
UN ECE REQUIREMENT: 7.2.2.1.3When the execution of an update may affect the safety of the vehicle, the vehicle manufacturer shall demonstrate how the update will be executed safely. This may be achieved through technical means and/or through a process that will require the vehicle user to provide verification that the vehicle is in a state where the update can be executed safely.

Erläuterung

Das Testprozedere kann in Abhängigkeit der betroffenen Systeme abgestuft werden, um das Testen effizienter zu gestalten. So können z. B. Infotainment-Systeme bereits während der Entwicklung ausreichend getestet werden, so dass nur die erfolgreiche Installation verifiziert werden muss. Während Safety-relevante Systeme auch im Fahrzeug getestet werden sollten, um auch ihre Funktionalität nach der Installation zu bestätigen. In einem solchen Fall kann z. B. getestet werden, ob sich die Aktuatoren ansprechen lassen oder zunächst ein Betatest im Feld durchgeführt wird, d. h., dass die neue Software parallel zu dem bisherigen Softwarestand läuft und die Ergebnisse des neuen Softwarestandes verifiziert werden.

OTA-I-050

Input: Testergebnis der Installation	
Das ***Fahrzeug*** sendet eine Bestätigung über die erfolgreiche Installation an die Fahrzeug-Datenbank / das Backend und an den Nutzer.	
R	Fahrzeug
A	Entwicklungsverantwortlicher
C	
I	Halter
Output: Bestätigung (z. B. aktualisierter Datenbankeintrag bzw. Meldung im IVI)	
Risiko-E047: Die Veränderung am Fahrzeug wird nicht dokumentiert, sodass es bei der Benutzung oder bei dem Weiterverkauf des Fahrzeuges zu einem negativen Kundenerlebnis kommt, da sich z. B. eine Funktion anders verhält als erwartet oder vorab des Updates. **Security-Risiko-E048:** Das erfolgreich durchgeführte Update wird abgestritten (z.B. kann dies bei Zusatz-Features zu einem Ausfall von Zahlungen führen). **Risiko-E049:** Durch eine nicht erfolgte Rückmeldung über den Erfolg des Updates kann der Verkauf einer Dienstleistung oder ein Rückruf nicht als abgeschlossen dokumentiert werden.	
UN ECE REQUIREMENT: 7.2.1.2.1Each RXSWIN shall be uniquely identifiable. When type approval relevant software is modified by the vehicle manufacturer, the RXSWIN shall be updated if it leads to a type approval extension or to a new type approval.	

7.2.2.4. After the execution of an update the vehicle manufacturer shall demonstrate how the following will be implemented: ▪ The vehicle user is able to be informed of the success (or failure) of the update; ▪ The vehicle user is able to be informed about the changes implemented and any related updates to the user manual (if applicable).

OTA-I-060

Input: Testergebnis der Installation	
Das ***Fahrzeug*** sendet eine Fehlermeldung über das fehlgeschlagene Update an die Fahrzeug-Datenbank / das Backend und ggf. an den Nutzer.	
R	Fahrzeug
A	Entwicklungsverantwortlicher
C	
I	Halter
Output: Fehlermeldung (z. B. aktualisierter Datenbankeintrag)	
Risiko-E050: Fehlermeldung wird nicht gesendet oder mit falschem Inhalt versendet, so dass der OEM nicht über aufgetretene Fehler informiert wird.	

OTA-I-070

Input: Fehlermeldung (z. B. aktualisierter Datenbankeintrag)	
Das ***Fahrzeug*** versucht einen sicheren Zustand wiederherzustellen und kommuniziert dies und das Ergebnis des Versuchs an den *Fahrer* / *Halter*.	
R	Fahrzeug
A	Update-Koordinator
C	
I	Halter, Entwicklungsverantwortlicher
Risiko-E051: Das Fahrzeug kann in keinen Fail/Safe Modus überführt werden, so dass das Fahrzeug ggf. für den Fahrer nicht mehr nutzbar ist. **Risiko-E051b**: Das Fahrzeug verliert Verbindung zum Backend und das Fahrzeug verbleibt im Privacy Mode – einem Modus bei dem bewusst die Kommunikation zum OEM durch den Kunden abgestellt wird. Ein Fail/Safe Modus mit ausreichender Funktionalität muss sichergestellt sein.	
Output: (ergänzte) Fehlermeldung	
UN ECE REQUIREMENT:	

7.2.2.1.1The vehicle manufacturer shall ensure that the vehicle is able to restore systems to their previous version in case of a failed or interrupted update or that the vehicle can be placed into a safe state after a failed or interrupted update.

Erläuterung

Ziel ist es, die Einschränkungen für den Kunden so gering zu möglich zu halten und nach einem Fehler möglichst alle Funktionalitäten wieder bereitzustellen. Zu überlegen ist, ob bei sicherheitsrelevanten Fehlern, deren Behebung fehlgeschlagen ist, die entsprechende Funktion aus Sicherheitsgründen zu deaktivieren ist. Minimalziel ist es, das Fahrzeug in einen fahrbereiten Zustand zu versetzen. Anschließend ist das Wiederherstellen des normalen Zustanden in der Werkstatt herbeizuführen.

OTA-I-080

Input: Bestätigung oder Fehlermeldung	
Der ***Digitale Dienstleister*** beobachtet den Fortschritt des Updateprozesses. Hierzu zählen die Überwachung des prozessualen Ablaufs während der gesamten Updateverteilung als auch die Überwachung der Installationsroutine und der Umgang / die Eskalation mit möglichen Fehlermeldungen einzelner Fahrzeuge.	
R	Digitaler Dienstleister
A	
C	Update-Koordinator
I	Entwicklungsverantwortlicher
Output: Status des gesamten Prozesses / Bug-Report	
Risiko-E052: (Systematische) Fehler im Update oder durch eine Installationsroutine können erst bei den Kunden im Feld auftreten, ohne dass eine direkte Betreuung durch einen Werkstattmitarbeiter möglich wäre. Hier kann es zu hohen Einschränkungen der Leistungsfähigkeit des Fahrzeugs gegenüber dem Kunden kommen.	

OTA-I-090

Input: Status des gesamten Prozesses / Bug-Report	
Der ***Update-Koordinator*** überwacht den Update-Prozess und koordiniert die Eskalation möglicher Fehler im Ablauf.	
R	Update-Koordinator
A	
C	Entwicklungsverantwortliche

I	Regulatorische Behörde
Output: Statusbericht (nur bei Rückrufen)	
Konditionen: (A) ggf. ist das Update im Werkstattprozess weiterzuverfolgen (→ WRK-V-010). **(B)** ggf. weist das Update einen systematischen Fehler auf. In diesem Fall muss die Verteilung gestoppt und ein aktualisiertes Update generiert werden (→ OTA-G-050). **(C)** Bei einem laufenden Rückruf leitet der *Update-Koordinator* in vereinbarten Zeitintervallen einen Statusbericht an die *Regulatorische Behörde* weiter.	

OTA-I-100 [nur bei reg. Beh.]

Input: Statusbericht	
Die ***Regulatorische Behörde*** aktualisiert die Rückruf-Datenbank und schließt gegebenenfalls den Rückruf entsprechend ihrer jeweiligen Vorgehensweise ab	
R	Regulatorische Behörde
A	
C	
I	Update-Koordinator
Output: Statusbericht	

4 SIMULATION

Zur detaillierten Analyse des Prozesses und Vertiefung der Inhalte wurde eine Simulation im Rahmen dieses Projektes für OTA-Updates programmiert. Diese Simulation ist dabei hauptsächlich „Mittel zum Zweck“, um Erkenntnisse, z. B. durch die Modellbildung einzelner Aktivitäten, zu generieren und den Prozess zu validieren.

4.1 MATLAB-Tool

Die Simulation wurde mittels MATLAB umgesetzt und besitzt den in Abbildung 12 dargestellten logischen Ablauf.

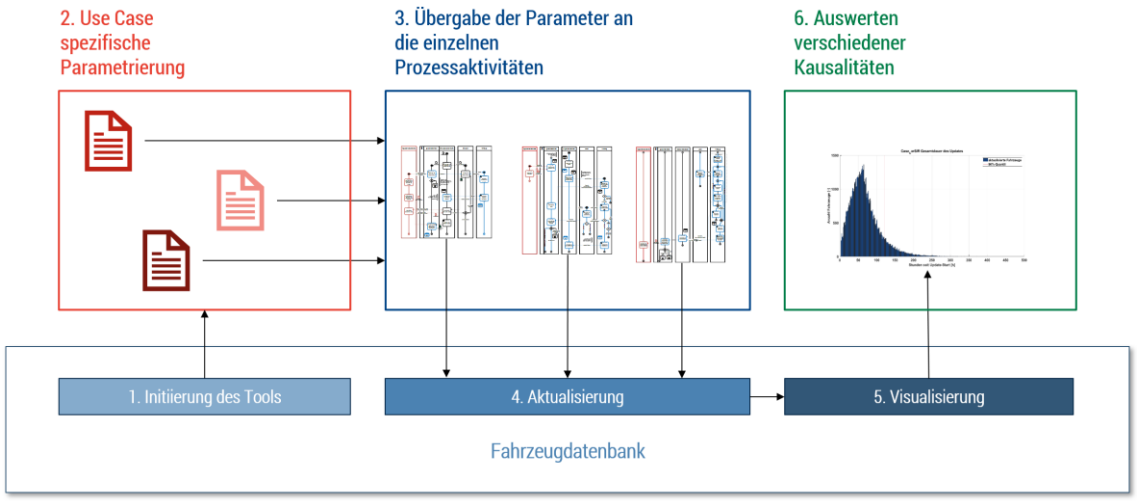


Abbildung 12: logischer Aufbau des Simulationswerkzeugs

Im Folgenden werden die einzelnen in der in Abbildung 12 visualisierten Schritte einzeln erläutert.

(1) Zunächst wird das Simulationswerkzeug initiiert. Hierzu zählt auch das Einlesen der Fahrzeugdaten. Diese Daten identifizieren die Zielfahrzeuge über das Kennzeichen und beschreiben die technische Ausstattung der Fahrzeuge hinsichtlich Leistungsfähigkeit der

Connectivity Unit und der Ziel ECU. Die entsprechende Datenbank ist die Basis, die auch um die Ergebnisse der Simulation ergänzt wird (repräsentiert durch die blaue, horizontale Box unten). Die wichtigste Dimension in der Simulation ist dabei die Dauer, die das Update über alle Prozessschritte benötigt (`duraTotal`).

(2) Im zweiten Schritt muss der Simulationsnutzer, den Prozess parametrisieren und ihn an die Rahmenbedingungen anpassen. Hierzu stehen 42 Parameter zur Verfügung, die

- die Technik (verbaute Mobilfunktechnik, Rechenleistung der ECU, etc.),
- die Beziehung zwischen Kunden und OEM (in welchen Abständen wird der Kunde informiert, wie häufig darf das Update abgelehnt werden, etc.) und
- den Kunden selbst (wie häufig lehnt der Kunde das Update ab, wann ist der Kunde am Fahrzeug, etc.)

beschreiben.

Vor allem um das Kundenverhalten realistischer abzubilden, wurden Verteilungsfunktionen in die Simulation integriert. Diese können ebenfalls parametrisiert werden, um so über den Schwerpunkt, die Streckung und die Dichte der Verteilung, die jeweilige Kundengruppe besser abzubilden.

(3) Das Tool übergibt diese Parameter nun schrittweise an die programmierten Aktivitäten des in Kapitel 3 dargestellten idealisierten OTA-Prozess, berechnet für jedes Fahrzeug in Abhängigkeit der Technik, des Fahrers und der Infrastruktur eine spezifische Durchlaufzeit für jede dieser Aktivitäten und summiert diese Werte zu einer Gesamtdurchlaufzeit auf.

Insgesamt wurden 13 der 26 obligatorischen Aktivitäten und zwei optionale Aktivitäten programmiert. Bei den weiteren Aktivitäten wurde kein kritischer Einfluss auf die Durchlaufzeit angenommen, da sie entweder parallel stattfinden oder in Parametern anderer Aktivitäten berücksichtigt werden können.

(4) Nach jeder Aktivität wird entsprechend die Fahrzeugdatenbank, um die Dauer des Updates je Aktivität und Fahrzeug und um den Erfolg / Misserfolg der Aktivität aktualisiert.

(5) In Vorbereitung auf die abschließende Analyse, werden die Daten aufbereitet und visualisiert.

(6) Anhand der Simulationsergebnisse können nun verschiedene Kausalitäten entlang der Prozesskette manuell analysiert werden. Dabei ist festzustellen, dass der Einfluss einiger Faktoren degressiv ist (entgegen der Erwartung eines linearen Einflusses) oder Abhängigkeiten zwischen Parametern existieren, wie im Folgenden gezeigt wird.

4.2 Use Case Vergleich

Für die Durchführung dieser Simulation wurden drei verschiedene Anwendungsfälle unterschieden, die jeweils eine signifikant unterschiedliche Parametrisierung nach sich ziehen. Nachfolgend sind die Anwendungsfälle zunächst tabellarisch beschrieben. Diese Informationen entsprechen den Informationen, einer initialen Updatebedarfsmeldung zugrunde liegen könnten, welche den OTA-Updateprozess initiiert.

Anzumerken ist, dass die Ergebnisse hier rein exemplarisch sind und für sich keine verbindlichen Aussagen treffen. Vielmehr werden Sie genutzt, um bei gleichen Grundannahmen die Auswirkung von Prozess- und Parametervariationen auf die Prozessperformance zu untersuchen. Für den Abgleich mit der Realität fehlten quantifizierbare Vergleichsdaten. Mit diesen hätte man im ersten Schritt einen durchgeführten Updateprozess abbilden können, um ihn anschließend zu optimieren. Weil dieses Abbild der Realität fehlte, mussten zum Teil begründete Annahmen getroffen wurden.

4.2.1 Car-Security-Incident-Response (Car-SIR)

Name	Update zur Beseitigung eines Security Problems im Bluetooth Stack.
Trigger	Sicherheitsforscher entdecken eine Schwachstelle in der Head Unit, sodass sich Schadsoftware auf dem Head Unit installieren lässt.
Technische Beschreibung	Durch das Vorhandensein eines Fehlers im Bluetooth Stack lässt sich Schadsoftware auf Head Unit eines Herstellers aufspielen. Die Head Unit ist an den System-CAN-Bus angeschlossen. Dieser setzt zum aktuellen Zeitpunkt keine dezidierten Sicherheitsmechanismen ein, sodass über die Head Unit andere Steuergeräte in den Diagnosemodus versetzt werden können und zum Download eines modifizierten Firmware Updates bewegt werden können. Über das modifizierte Firmware-Updates lassen sich Sicherheitskritischen CAN-Signale erzeugen, sodass kritische Fahrzeugfunktionalität beeinflusst wird.
Betroffene Software/ECUs	Head Unit mit fehlerhaftem Bluetooth Stack (Blueborn Lücke) Head Unit mit Schadsoftware Evtl. Steuergeräte mit modifizierter Firmware
Maßnahmen	Softwareupdate auf der Head Unit, sodass ein fehlerbereinigter Bluetooth Stack verwendet wird. Ggfs. entfernen von Schadcode auf der Head Unit (ggfs. die gesamte Software neu aufspielen) Ggfs. aktualisieren der kompromittierten Firmware auf anderen Steuergeräten
Rechtliche Rahmenbedingungen	Sicherheitslücken, die dazu führen können, dass kritische Funktionalität im Fahrzeug beeinträchtigt wird, sind meldepflichtig und rückrufrelevant.

Zunächst wurde der Use Case Car-SIR initial parametrisiert. Hierbei wurde von einem zeitkritischen, verpflichtenden Security-Update von 10 MB Größe ausgegangen. Dieses benötigte rund 16 Stunden bis zur Bereitstellung im CDN und wurde sowohl über WiFi als auch Mobilfunk an 100.000 Fahrzeuge per Shoulder-Tap Methode verteilt. Das Zielsystem, welches das Update adressiert, ist die Head-Unit, wodurch von geringen Einschränkungen für den Endnutzer ausgegangen wird, da eine hohe Rechnerleistung zur Verfügung steht und kein primär Safety-relevantes System adressiert wird. Dies wurde in der Simulation durch eine höhere Wahrscheinlichkeit für die Genehmigung des Updates durch den Kunden berücksichtigt.

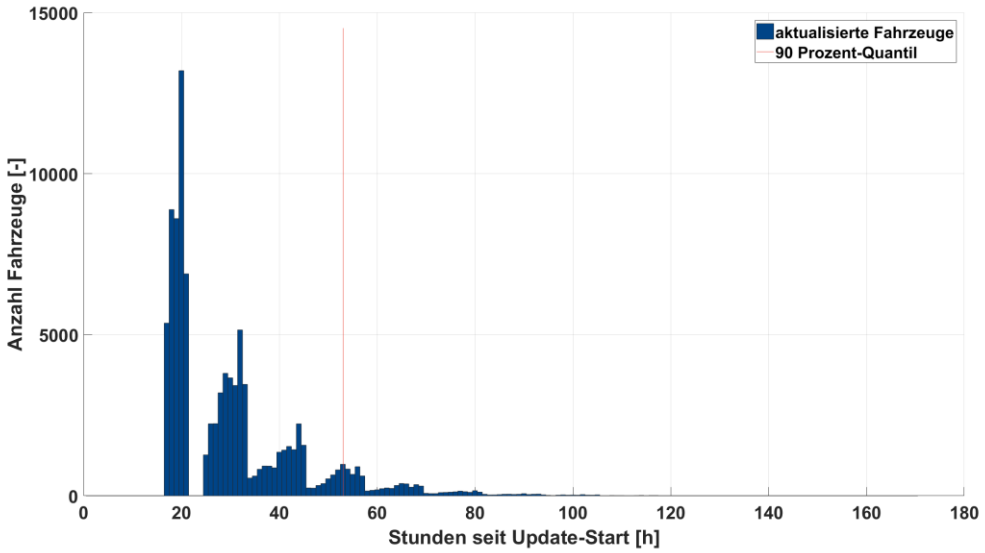


Abbildung 13: Use Case Car-SIR - Durchlaufzeit je Fahrzeug

Abbildung 13 zeigt das 90 % der Fahrzeuge nach rund 55 Stunden aktualisiert waren. Dennoch sind vereinzelt Fahrzeuge erst nach 170 Stunden und damit nach mehr als der dreifachen Zeit aktualisiert worden. In der Grafik sind nur die erfolgreichen Fahrzeuge berücksichtigt. Letztlich sind bei diesem Update rund 3,3 % der Updates fehlgeschlagen, wie Abbildung 14 zeigt.

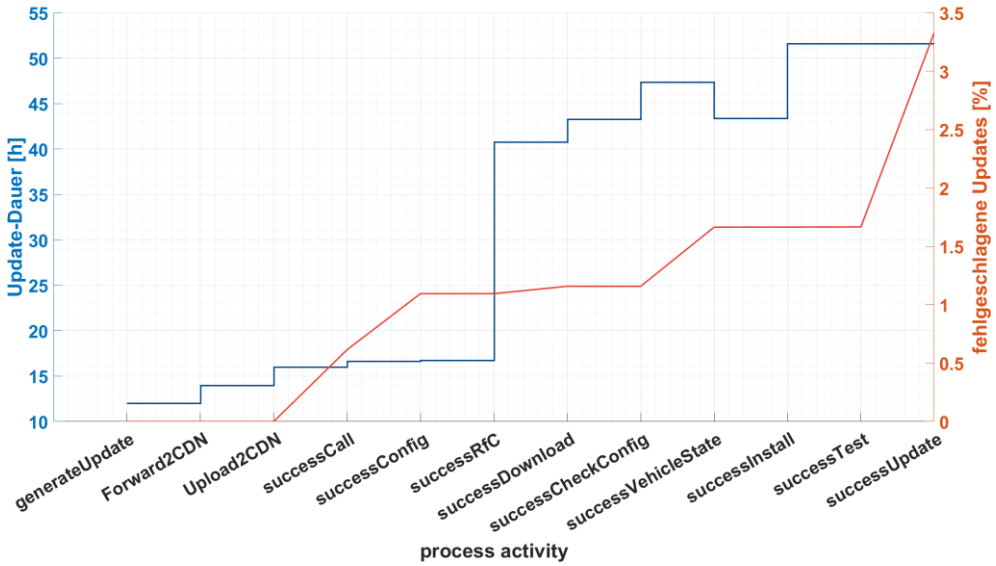


Abbildung 14: Use Case Car-SIR - Durchlaufzeit je Aktivität

Außerdem zeigt diese Grafik die Dauer der einzelnen Aktivitäten im Verhältnis zueinander. Je höher eine Stufe zwischen zwei Aktivitäten (process acitivity) ist, desto länger dauert der Prozessschritt. Aufgrund der geringen Updategröße und der Übertragung per Mobilfunk sind die Aktivitäten keine Engpässe, die die Übertragung oder die automatische, technische Verarbeitung des Updates beschreiben. Vielmehr ist es die Bestätigung der Installation durch den Kunden, die die Durchlaufzeit verzögert (*successRfC*). Dies kann mehrere Gründe haben: zum einen kann es schlicht sein, dass aufgrund der kurzen Gesamtdurchlaufzeit die Nutzer schlicht nicht am Fahrzeug waren oder zum anderen, dass das Update zu einem ungünstigen Zeitpunkt vorgeschlagen wurde.

Zum Meilenstein *successVehicleState* gibt es einen logischen Fehler in der Darstellung, da hier die Updatedauer sinkt. Dies liegt an dem Anstieg an fehlgeschlagenen Updates zu dieser Aktivität. Diese entfallen im folgenden Schritt in der Berechnung, da die Datenbasis der Abbildung nur auf den Werten erfolgreicher Fahrzeuge basiert. Zu diesen fehlgeschlagenen Fahrzeugen zählen wohl Fahrzeuge mit bereits auffällig höherer Updatedauer als der Durchschnitt. Entfallen diese nun, sinkt durch die fehlenden Ausreißer mit sehr hoher Updatedauer der Durchschnitt der Updatedauer.

4.2.2 Aktualisieren der Software im Infotainment

Name	Aktualisieren der Software im Infotainment
Trigger	Ein Automobilhersteller aktualisiert sein Infotainment-System auf eine neue Hauptversion. Ausgangspunkt war Kundenfeedback für eine einfachere Bedienung, die Verbesserung der Effizienz und Kompatibilität mit Dritten.
Technische Beschreibung	Im Vergleich zur Konkurrenz empfinden die Kunden das System als „träge“ und ihnen fehlen wichtige Funktionen. Außerdem sind Smartphones mit der neuesten Betriebssystemversion (z. B. aufgrund eines neuen Übertragungsstandards) nicht mehr kompatibel.
Betroffene Software/ECUs	Infotainment System
Betroffene Fahrzeuge /Fahrzeugklassen	Mehrere Mittelklasse- und Oberklassefahrzeuge eines Herstellers
Maßnahmen	Geändert wird einerseits die Menüführung, neue Funktionalitäten werden hinzugefügt, die Software wird „aufgeräumt“ und an die neuesten Versionen von Drittanbietern, wie Apple, Android oder Spotify, angepasst.
Rechtliche Rahmenbedingungen	

Aufgrund der Systemarchitektur und dem Umfang der neuen Funktionen wurde ein Update-Paket mit einer Größe von 1000 MB simuliert. Da das Update einen Mehrwert für den Kunden bereitstellt und es keine Fehlerbehebung ist, wird auf eine Übertragung auf Kosten des OEMs verzichtet und es wird ein hoher Anteil an Übertragungen durch WiFi oder USB-Tethering angenommen (70 %). Die weiteren 30 % werden über Mobilfunk durch die Fahrzeuge geladen, deren Besitzer eine entsprechende Mobilfunk-Option abgeschlossen haben, die als Sonderausstattung für die beschriebenen Mittelklassefahrzeuge zur Verfügung steht.

Durch die Updategröße und den hohen Anteil der WiFi-Übertragungen sinkt jedoch, die Wahrscheinlichkeit, dass das Fahrzeug erreicht wird und das Update in einer Sitzung erfolgreich und komplett heruntergeladen wird. Zudem verzögert sich das Erreichen der Kunden durch die zum Teil fehlende Mobilfunk-Verbindung, da zusätzlich weitere Kanäle zur Kommunikation des Updates an den Kunden gewählt werden müssen (z. B. Newsletter, Apps, etc.).

Da von vornherein von einer solchen Verzögerung ausgegangen werden kann und aufgrund der nachrangigen Priorität des Updates, wurde das Effizienzmaximum für die Server so ausgewählt, dass eine entsprechende, geringe Anzahl an Serverslots bereitgestellt wurde. Zudem wurde eine hohe Wahrscheinlichkeit für das Vorhandensein einer korrekten Fahrzeug-Konfiguration angenommen (*pSuccessConfig*), da es keine Interdependenzen des Infotainmentsystems mit anderen ECU bezüglich der Installation gibt. Hierdurch – zumal das Update nicht Safety-relevant

ist – gibt es auch geringere Anforderungen an den Fahrzeugzustand, der hergestellt werden muss. Die geringeren Anforderungen sind in der Simulation als höhere Wahrscheinlichkeit berücksichtigt, dass der Nutzer / das Fahrzeug diesen Zustand erfolgreich herstellen (*pSuccessVehicleState*).

Die Wahrscheinlichkeit, dass das Update angenommen wird (*pSuccessRfC*), ist eher gering, da das Update einerseits nicht kritisch ist (es behebt keinen Fehler) gleichzeitig aber einen höheren Aufwand für den Kunden darstellt. Die Installations- und Downloaddauer ist aufgrund der Paketgröße hoch. Dieser Effekt gleicht den Nachteil des geringen Downloaderfolges etwas aus, da bis zur Autorisierung des Updates der Download mehrfach gestartet / fortgesetzt werden kann. Wenn das Update letztlich autorisiert wird, kann davon ausgegangen werden, dass der Download schon erfolgt ist. Anschließend kann es jedoch aufgrund der Länge des Installationsvorgangs zu Abbrüchen des Installationsvorgangs kommen, sollte das Fahrzeug in der Zwischenzeit bewegt werden (*pSuccessInstall*).

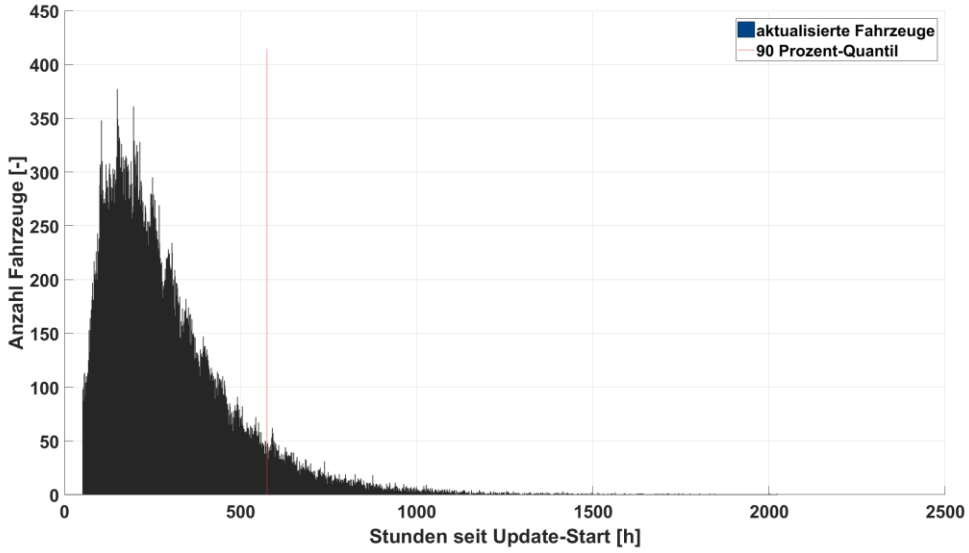


Abbildung 15: Use Case Infotainment - Durchlaufzeit je Fahrzeug

Zur Erreichung des 90 Prozent-Quantils dauert es bei der gleichen Anzahl an Fahrzeugen in diesem Use Case bereits rund 550 Stunden. Während im Use Case Car-SIR rund 10.000 Fahrzeuge in der Anfangsphase pro Stunde aktualisiert wurden, sind es hier nur maximal 375 Fahrzeuge.

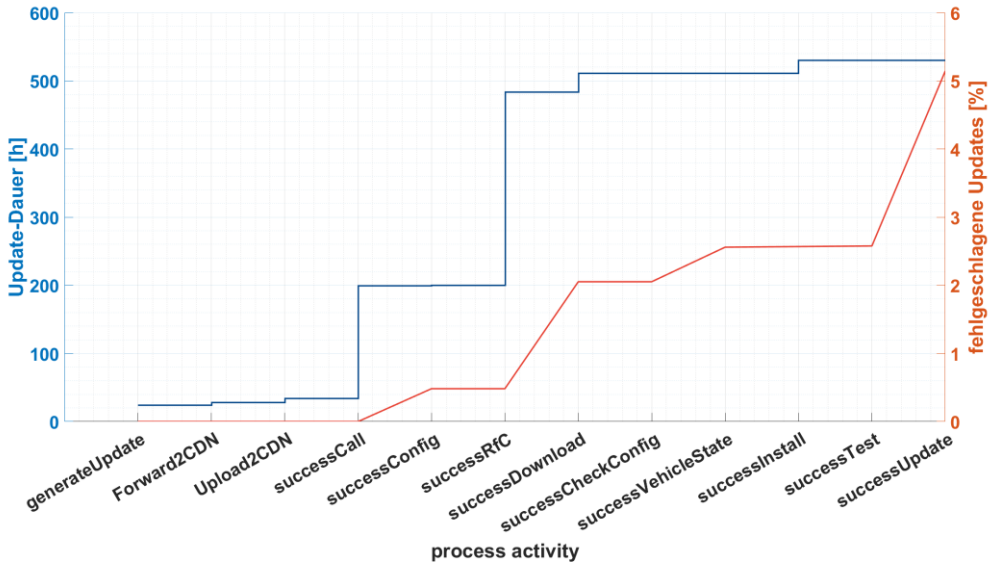


Abbildung 16: Use Case Infotainment - Durchlaufzeit je Aktivität

Die meiste Zeit während des Update-Prozess vergeht beim Benachrichtigen des Fahrzeugs bzw. des Kunden, wegen des hohen Anteils an WiFi-Übertragungen. Anschließend vergeht viel Zeit bis das Update vom Kunden autorisiert wird. Der eigentliche Download selbst trägt zur Gesamtlaufzeit verhältnismäßig wenig bei.

4.2.3 Rückruf einer Software im Feld auf Anweisung des KBA

Name	Rückruf zur Fehlerbeseitigung im Airbag
Trigger	Ein Automobilhersteller stellt fest, dass ein Softwaremodul im Sensor eines Zulieferers fehlerhaft ist und es so zu unter gewissen Rahmenbedingungen zu Verzögerungen beim Auslösen des Seitenairbags kommen kann.
Technische Beschreibung	Die Software des Seitenairbag-Sensors (Beschleunigung) ist fehlerhaft, sodass bei bestimmten Randbedingungen Daten erst spät bereitgestellt werden, sodass es im Falle eines Unfalls zu einer Verzögerung bei der Auslösung des Seitenairbags kommt.
Betroffene Software/ECUs	Seitenairbag-Sensor (Beschleunigung)
Betroffene Fahrzeuge /Fahrzeugklassen	Mehrere Mittelklassefahrzeuge mehrerer Hersteller
Maßnahmen	Fehlerbeseitigung im Softwaremodul durch den Zulieferer und aufbringen der fehlerbereinigten Software durch den OEM.
Rechtliche Rahmenbedingungen	Airbags gehören zum Sicherheitssystem (Insassenschutz) im Fahrzeug. ECE R114 regelt den Austausch von Airbag-Systemen

Der dritte Use Case beschreibt einen Rückruf im Feld aufgrund eines fehlerhaften Seitenairbag-Sensors. Dies ist ein Safety-relevantes, Behörden-relevantes, zeitkritisches und obligatorisches Update, für das fiktiv eine Frist von sieben Tagen bis zum Ende der OTA-Aktualisierung festgesetzt wurde.

Das Update-Paket ist 10 MB groß, wird zu 80 % per Mobilfunk und via Shoulder-Tap verteilt. Aufgrund der Dringlichkeit wird versucht, das Fahrzeug alle 6 Stunden zu erreichen – bei entsprechend hohen Serverkapazitäten. Dabei wird angenommen, dass dadurch die Wahrscheinlichkeit sinkt, dass das Fahrzeug zu jeder dieser Anfragen auch erreicht wird. Zusätzlich wird angenommen, dass die Konfiguration häufiger fehlerhaft ist, da davon ausgegangen werden kann, dass der Airbag oder das Steuergerät nach einem Unfall ausgetauscht und dies nicht sauber kommuniziert / digitalisiert wurde. Dementgegen wurden für angenommen für die Annahme des Updates, die Herstellung des richtigen Zustandes für die Installation und eine erfolgreiche Installation hohe Wahrscheinlichkeiten angesetzt. Einerseits weil der Nutzer selbst gefährdet sein könnte, wenn er das Update nicht installiert und andererseits, weil nur eine einzelne ECU betroffen ist, so dass die Installation selbst keine hohen Anforderungen hat.

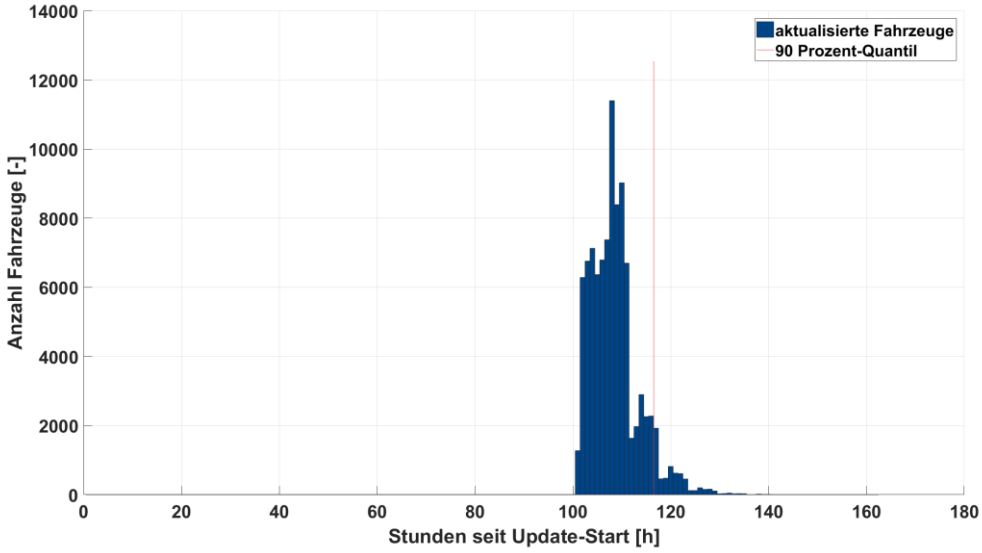


Abbildung 17: Use Case Airbag - Durchlaufzeit je Fahrzeug

In diesem Beispiel ist eine lange Vorlaufszeit abgebildet, die aufgrund der Komplexität der Lieferantenbeziehungen entstanden ist. Software und Sensor werden nicht vom OEM oder Tier-1 Zulieferer hergestellt, sondern von einer Zuliefererfirma, die in keinem direkten Vertragsverhältnis mit dem OEM steht geliefert. Nach der Bedarfsmeldung wird die Problematik in der Lieferkette eskaliert. Bei den hier getroffenen Annahmen ist dennoch eine Verteilung an einen Großteil der Fahrzeuge innerhalb der fiktiven sieben Tagefrist möglich.

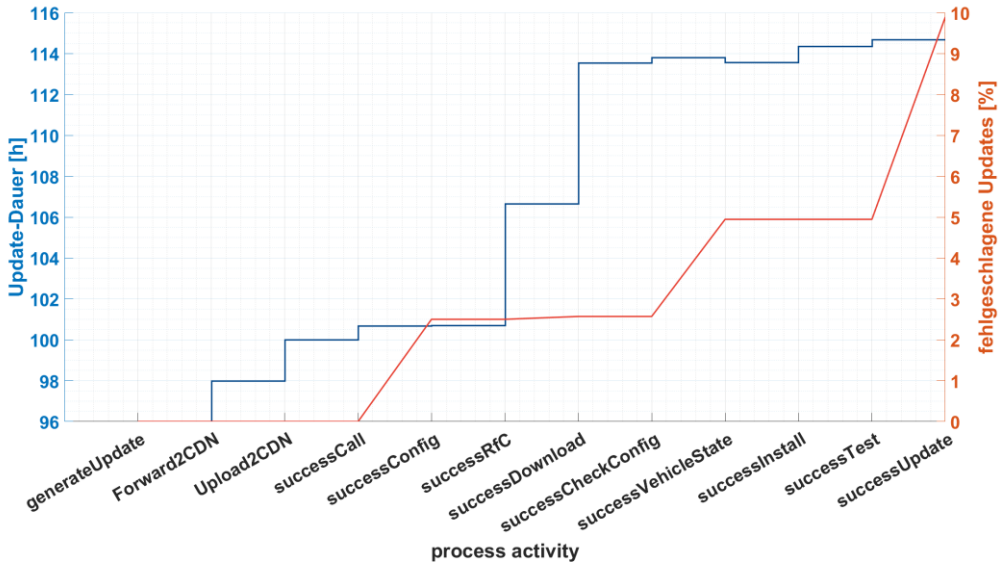


Abbildung 18: Use Case Infotainment - Durchlaufzeit je Aktivität

Nachteilig ist jedoch eine hohe Anzahl an fehlgeschlagenen Updates, wie in Abbildung 18 zu sehen ist, die sich mit der fehlerhaften Konfiguration, dem falschen Fahrzeugzustand oder einer unzureichenden Überprüfung der Installation begründen. Darüber hinaus ist in diesem Use Case zu erkennen, dass diesmal der Erfolg des Downloads einen signifikanten Einfluss auf die Durchlaufzeit hat.

4.2.4 Schlussfolgerung

Alle drei dargestellten Use Cases sind exemplarisch und nicht anhand von realen Daten simuliert. Dennoch sind logische Annahmen getroffen, um mit der Simulation die Use Cases zu vergleichen.

Festzuhalten ist, dass in allen drei Fällen die Bestätigung des Updates durch den Kunden einen signifikanten Einfluss hat. Die Gestaltung der Benachrichtigung über das Update ist somit ein wichtiger Faktor, um die Durchlaufzeit des OTA-Updates zu optimieren. Darüber hinaus sind weitere wichtige Aktivitäten das Senden der Nachricht an das Fahrzeug selbst, der Downloadprozess und die Konfigurationsprüfung. Der Einfluss der einzelnen Parameter wird im folgenden Abschnitt untersucht.

4.3 Prozessoptimierung

Mittels des Simulationstools wurde iterativ der OTA-Prozess für den Use Case „Car-Security-Incident-Response (Car-SIR)" optimiert. Schrittweise wurde der Einfluss jeder Parameter-Variation auf das Gesamtergebnis betrachtet. Das ideelle Ziel bei einer Sicherheitslücke muss es sein, innerhalb von 24 Stunden mit einem Security-relevanten Patch zu reagieren und die Lücke zu schließen. In Abschnitt 4.2.1 wurde dieser Use Case initial parametriet, dabei betrug die Durchlaufzeit rund **216 Stunden oder 9 Tage**.

Von diesem initialen Use Case ausgehend, wurden nun schrittweise die einzelnen Parameter variiert, um deren Einfluss auf die Gesamtzeit zu untersuchen. Dabei wird jeweils nur ein Parameter zur selben Zeit variiert. Anschließend wird für diesen Parameter der effizienteste Wert ausgewählt. Mit diesem Optimum wird der Prozess neuberechnet und auf Basis dieses Zwischenergebnisses der nächste Parameter variiert.

Zielkenngröße ist jeweils das 90%-Quantil der Gesamt-Updatedauer, d. h. der Zeitpunkt zu dem 90% der Fahrzeuge erfolgreich aktualisiert sind. Für das Ausgangsszenario beträgt dieser Wert rund **216 Stunden oder 9 Tage**.

4.3.1 Statische Einflussfaktoren

Einige Parameter tragen als konstante Faktoren zur Dauer des Update-Prozesses bei. Sie wurden trotzdem zur Simulation hinzugefügt, um auf Optimierungspotentiale innerhalb dieser Aktivitäten aufmerksam zu machen. Dies kann durch interne Effizienzsteigerungen erreicht werden oder durch Verbesserungen in der Kommunikation zwischen Stakeholdern. Zu diesen Einflussfaktoren zählen folgende Parameter:

- Dauer zum Generieren des Updates,
- Dauer zum Kommunizieren des Updates an den Digitalen Dienstleister,
- Dauer zum Hochladen des Updates in ein CDN, etc.

Darüber hinaus wurden weitere technische, konstante Einflussfaktoren implementiert, die zum Teil einen Einfluss auf die Dauer haben, die das Fahrzeug nicht für den Kunden zur Verfügung stehen, z. B. während der Installation:

- Dauer zum Verarbeiten einer Serveranfrage,
- Dauer zum Überprüfen / Validieren des Updates vor der Installation,
- Dauer zum Herstellen des richtigen Fahrzeugzustandes,
- Dauer zum Überprüfen des Erfolgs der Installationsroutine, etc.

4.3.2 Dynamische Einflussfaktoren

Darüber hinaus gibt es Einflussfaktoren, die für jedes Update durch den Update-Koordinator variiert werden können. Diese dynamischen Faktoren und der Einfluss ihrer Variation werden im Folgenden untersucht.

Anzahl Server-Slots / Anzahl Fahrzeuge

Zunächst wurde die Anzahl der Fahrzeuge variiert und dabei auch die Anzahl verfügbarer Server-Slots berücksichtigt.

Bleibt diese Anzahl der Server-Slots absolut konstant, steigt die Dauer des Updates mit steigender Anzahl der Fahrzeuge.

Hält man die Anzahl der Fahrzeuge konstant und variiert die Anzahl der Server Slots zeigt sich ein degressiv abnehmender Verlauf. Der unabhängig von den absoluten Größen ist, sondern sich nach dem Verhältnis von der Anzahl der Server-Slots zu der Anzahl der Fahrzeuge – siehe Abbildung 19.

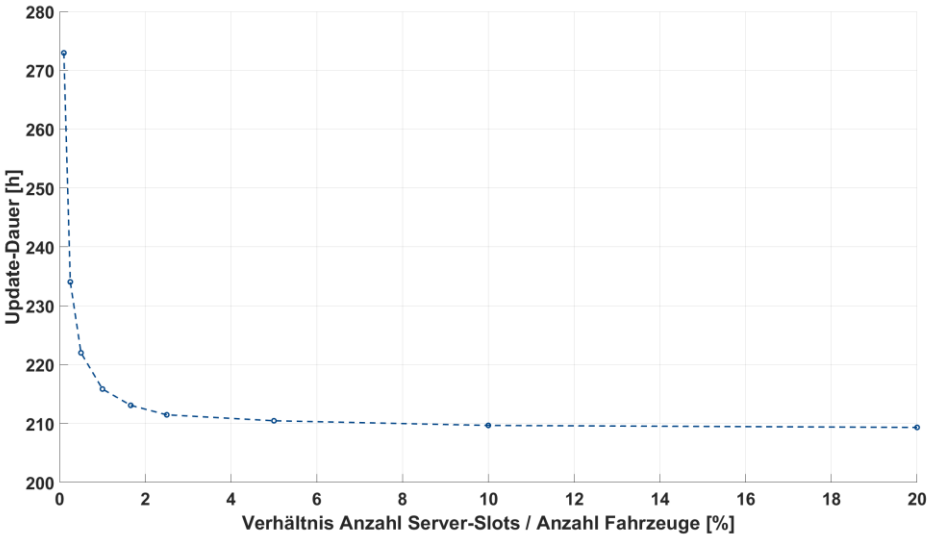


Abbildung 19: Auswertung Anzahl Server-Slots / Anzahl Fahrzeuge

Das heißt, dass eine immer weitere Steigerung der Server-Kapazitäten letztlich nur noch einen minimalen Einfluss auf die Durchlaufzeit des Updates hat. Angenommen, dass steigende Kapazitäten auch steigende Kosten verursachen, ist dies nicht mehr wirtschaftlich sobald die Kurve nahezu horizontal verläuft. Aus Effizienzsicht ist das Optimum in der Beuge zu suchen. Hier beginnt die Kurve immer langsamer zu sinken und jede Investition in weitere Kapazitäten hat einen immer geringer werdenden Effekt auf die Update-Dauer.

Für die Optimierung dieses Use Cases wird in Relation zur Update-Größe und Dringlichkeit eine letztlich noch effiziente aber auch möglichst effektive Verbreitung des Updates avisiert und deshalb ein Verhältnis von 2,5 Prozentpunkten (vorher 1 Prozentpunkt) am Ende der Beuge für

die weiteren Analysen ausgewählt, da ab hier der horizontale Verlauf beginnt. Zur Optimierung der Rechenzeit wird mit 100.000 Fahrzeuge gerechnet, da – wie gezeigt – nur das relative Verhältnis beider Kennwerte und nicht die absoluten Werte für die Dauer ausschlaggebend sind.

Mit dieser Anpassung sinkt die Durchführungszeit für das gesamte Update von 216 Stunden auf 211,5 Stunden.

Die Zuteilung von Kapazitäten zu Updates und damit deren Priorisierung kann noch einen weiteren Einfluss haben, falls über eine Infrastruktur mehrere Updates verteilt werden. Dies wurde innerhalb dieses Projektes jedoch nicht betrachtet.

Verteilung der Übertragungstechnik

In der Simulation wird für jedes Fahrzeug anhand einer vorher festgelegten Verteilung bestimmt, mit welcher Übertragungstechnik das Update auf das Auto aufgespielt wird. Hierzu zählen die Übertragung mittels WiFi, 3G oder 4G. Alternativ kann dies auch per Tethering stattfinden, d. h., dass der Kunde das Update auf einen Datenträger speichert und dann diesen im Auto anschließt oder das er sein Smartphone als mobilen Hotspot bereitstellt, über den das Fahrzeug das Update herunterladen kann.

Variiert man diese Verteilung zeigt sich – bezogen auf das jeweilige Delta zwischen den Gesamtdurchlaufzeiten –, dass sich die Übertragungsgeschwindigkeit degressiv auf die Gesamtlaufzeit auswirkt. Bei Halbierung der Übertragungsgeschwindigkeit verdoppelt sich das Delta nicht, sondern steigt nur gedämpft – siehe Abbildung 20.

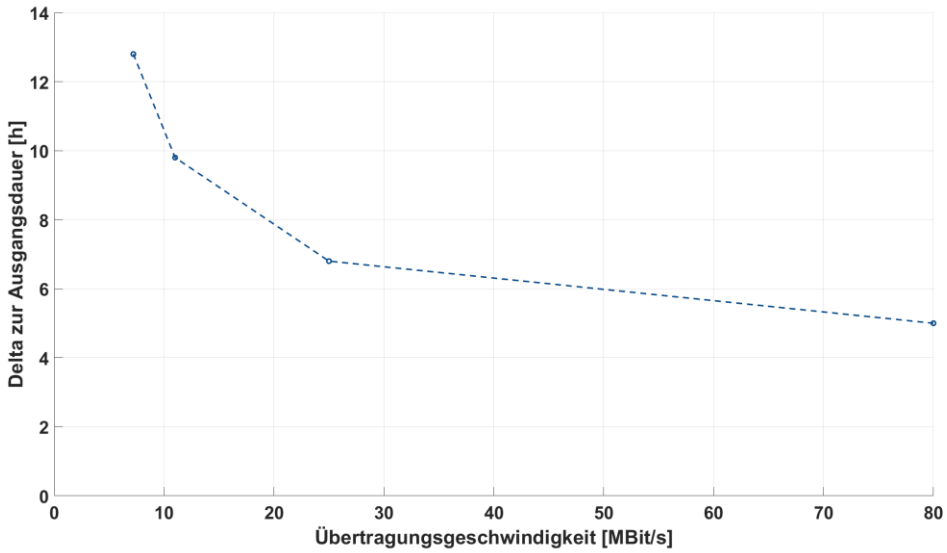


Abbildung 20: Auswertung der Verteilung der Übertragungstechnik

Bei Variation zusätzlich der Update-Größe ist ein ähnlicher Effekt zu beobachten. Das Delta zur Ausgangsdauer wird entsprechend für jede Übertragungsgeschwindigkeit, um einen Faktor erhöht. Es findet eine Parallelverschiebung der in Abbildung 20 dargestellten Kurve statt.

Hier überwiegen Abhängigkeiten zu anderen Parametern. Eine isolierte Betrachtung ist hier nicht ausreichend ist. So kann man davon ausgehen, dass Übertragungen großer Pakete via WiFi wahrscheinlicher abbrechen oder unterbrochen werden, da der Kunde nicht zwangsläufig bis zum Ende des Downloads sicherstellt, dass die Verbindung zum WiFi aufrecht erhalten bleibt. Während via Mobilfunk auch ein fahrendes Fahrzeug adressiert werden kann und der Erfolg des Downloads durch die vielfach höhere Netzabdeckung sichergestellt wird.

Für den weiteren Verlauf der Analyse wird der Anteil von WiFi und Tethering auf jeweils 10 % gesetzt, da aufgrund der Kritikalität das 10 MB große Update via Mobilfunk (auch auf eigene Kosten des Herstellers) verteilt wird. Die Annahme ist jedoch, dass einige Kunden sich dennoch im WiFi befinden, bzw. einen privaten Modus aktiviert haben und so keine Funkverbindung zwischen Fahrzeug und Backend zulassen. Aufgrund der regionalen Verteilung der Zielfahrzeuge wird davon ausgegangen, dass weitere 30 % das Update via 3G und die übrigen 50 % der Zielfahrzeuge das Update via 4G erhalten. Die Gesamtdurchlaufzeit verbleibt dabei bei 211,5 Stunden.

Größe des Updatepaketes

Im Folgenden wird nun die Größe des Update-Paketes von 10 MB über 100, 250, 500, 1000 auf 2500 MB angehoben.

Erwartungsgemäß wächst das Delta zum Ausgangsfall von 10 MB linear mit der Größe des Update-Paketes an. Das heißt, während die Gesamtdauer bei 250 MB um 2,75 Stunden ansteigt, steigt sie bei 2500 MB um 25,65 Stunden an. Dieser Effekt ist auch in Abhängigkeit des Verhältnisses Anzahl Server-Slots / Anzahl Fahrzeuge zu sehen. Hier variiert das absolute Delta bezüglich der Performance des gewählten Verhältnisses.

Zu berücksichtigen ist, dass von der Größe des Paketes die Installationszeit des einzelnen Nutzers stark variieren kann, wenn größere Updates das Fahrzeug länger außer Betrieb lassen, um die Installation durchzuführen. So kann sich ein größeres Update negativ auf die Wahrscheinlichkeit auswirken, dass ein Kunde ein Update annimmt. Wird das Zeitfenster für die Installation größer, wird es für den Kunden schwieriger spontan ein Update durchzuführen, es wird mehr Planung investiert, damit das Fahrzeug nicht während des Updatevorgangs benötigt wird.

Kommunikationsmethode (Shoulder-Tap bzw. Pull-Method)

Die Shoulder-Tap Methode ermöglicht eine zeitnahe Kommunikation des Updates an die Fahrzeugflotte. Dennoch können Effekte wie die fehlende Verbindung von Fahrzeugen diesen Vorteil reduzieren. Die Pull-Methode hat dahingegen den Vorteil einer besseren Planbarkeit. So können nicht nur Kosten gespart werden, weil weniger Serverkapazitäten zu einem Zeitraum benötigt werden, sondern es kann auch besser gegengesteuert falls das Update einen systematischen Fehler enthält.

Durch die verzögerte Verteilung steht hier mehr Reaktionszeit zur Verfügung, um das Update zu stoppen bevor zu viele Fahrzeuge betroffen sind. Hinzu kommt, dass bei der Pull-Methode das Fahrzeug selbst den Zeitpunkt auswählt, wenn es eine Verbindung zum Server hat. Dies kann

auch durch lernende Algorithmen unterstützt werden, um die Wahrscheinlichkeit für einen erfolgreichen Download und eine erfolgreiche Installation zu steigern.

Der Shoulder-Tap empfiehlt sich bei zeitkritischen Updates, ansonsten ist jedoch die Pull-Methode kosteneffizienter. Weshalb hier die Shoulder-Tap Methode für das Beispiel-Update gewählt wird. Hierfür wird eine Erreichbarkeit der Fahrzeuge von 90 % angenommen.

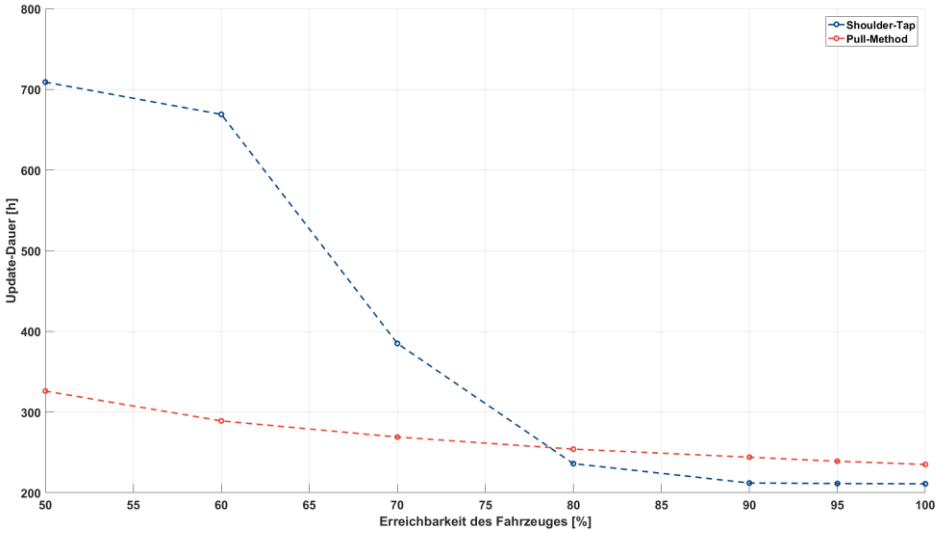


Abbildung 21: Vergleich beider Kommunikationsmethoden

Der Vergleich der beiden Methoden in Abhängigkeit der Erreichbarkeit der Fahrzeuge (Abbildung 21) zeigt, dass sich der Vorteil der Shoulder-Tap im Bereich zwischen 75 % und 80 % Erreichbarkeit egalisiert und sogar zu einem Nachteil für die Update-Dauer entwickelt. In der Simulation ist dies dadurch begründet, dass nach mehrfachem Nicht-Erreichen des Fahrzeuges, die Anfrage für mehrere Stunden zurückgestellt wird. Bei der Pull-Methode „weiß“ jedoch das Fahrzeug, wann es erreichbar ist und so steigt der Erfolg der einzelnen Anfragen.

Prüfen der richtigen Konfiguration

Die Prüfung der richtigen Konfiguration findet zu zwei Zeitpunkten in diesem Szenario statt. Zunächst während der Identifikation der Zielfahrzeuge und anschließend als Teil der Installationsroutine. Letztlich ist dies jedoch nur ein Korrekturfaktor in der Gesamtbetrachtung. Die Durchlaufzeit für die Fahrzeuge verändert sich nicht, da das Update bei nicht passenden Konfigurationen abbricht. Diese fehlgeschlagenen Fahrzeuge verzerren jedoch das 90%-Quantil, da aufgrund der Verteilung der Ausfall vor allem Fahrzeuge mit kürzere Durchlaufzeit betrifft, verschiebt sich das 90%-Quantil nach hinten. Einzelne Ausreißer mit sehr hohen Laufzeiten, fallen mehr ins Gewicht.

Die Wahrscheinlichkeit, die bei diesem Test hinterlegt wird, kann z. B. auf Erfahrungswerten basieren. Sie steigert sich mit besser integrierten Prozessen bis hin zur Werkstatt oder der verbesserten Kommunikation mit den Fahrzeugen.

Auch wenn dieser Faktor nur korrigierend eingreift, dient er trotzdem der Prognose der Updatedauer vor allem hinsichtlich möglicher Ausfallrisiken. Fallen mehr Fahrzeuge während des Updateprozesses aus, müssen diese Fahrzeuge ggf. in die Werkstatt zurückgerufen werden. Die Prozesskosten steigen und können so eingeplant werden.

Bestätigung des Updates

Die Aktivität des Genehmigens des Updates durch den Halter und z. B. ausführen des Updates im Fahrzeug über das IVI hat einen signifikanten Einfluss auf die Prozesskette.

Dabei sind vor allem zwei Parameter zu berücksichtigen. Die Regelmäßigkeit der Anfrage (z. B. im Fahrzeug durch das IVI) nach der Bestätigung und die Wahrscheinlichkeit dafür, dass der Fahrer das Update annimmt.

Der Einfluss der Anfrage auf die Update-Dauer ist dabei linear. Von rund 40 Stunden Update-Dauer bei einer Benachrichtigung alle 4 Stunden entwickelt sich die Kurve annähernd linear bis zum letzten getesteten Wert und einer Benachrichtigung alle 48 Stunden mit einer Update-Dauer von ungefähr 450 Stunden. Nimmt man an, dass es sich um ein Pflichtupdate handeln könnte, würde in diesem Szenario die Update-Dauer auf rund 21 Stunden sinken.

Der Einfluss der Anfrage auf die Update-Dauer ist dabei linear. Von rund 40 Stunden Update-Dauer bei einer Benachrichtigung alle 4 Stunden entwickelt sich die Kurve annähernd linear bis zum letzten getesteten Wert und einer Benachrichtigung alle 48 Stunden mit einer Update-Dauer von ungefähr 450 Stunden. Nimmt man an, dass es sich um ein Pflichtupdate handeln könnte, würde in diesem Szenario die Update-Dauer auf rund 21 Stunden sinken (Dauer bis zur nächsten Anfrage gleich 0). Die Werte wurden mit einer Wahrscheinlichkeit für die Annahme des Updates von 25 % aufgenommen.

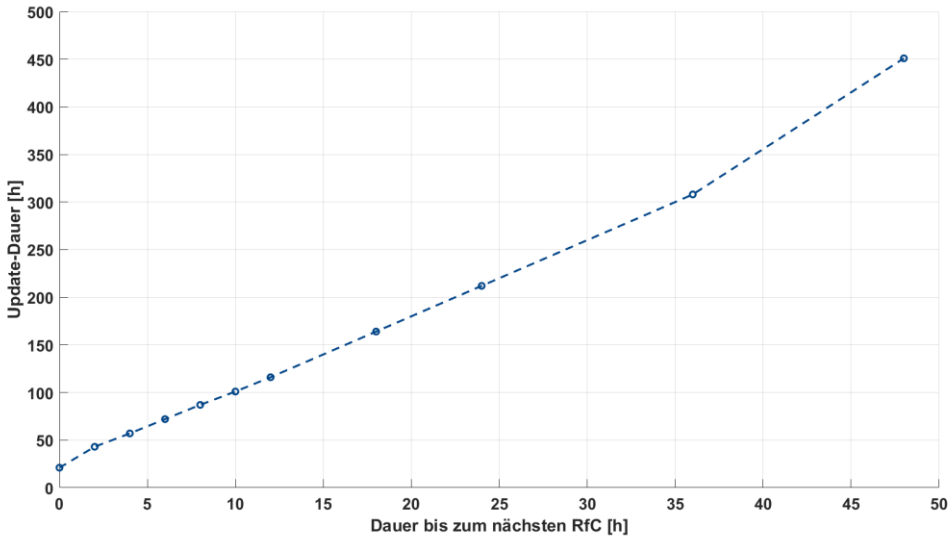


Abbildung 22: Einfluss der Anfrage nach Bestätigung beim Kunden

Zu untersuchen ist, inwiefern die Häufigkeit der Benachrichtigung eventuell einen Einfluss auf die Bereitschaft des Kunden hat, das Update auch tatsächlich zu bestätigen. Im Folgenden wird angenommen, dass ein häufiges Benachrichtigen des Kunden dazu führt, dass der Fahrer zu häufig in unpassenden Situationen konfrontiert wird und er deshalb routinemäßig die Nachricht ignoriert. Hierdurch müsste also die Wahrscheinlichkeit sinken, dass der Kunde das Update bei der Erinnerung bestätigt. Dies wird im nächsten Schritt untersucht.

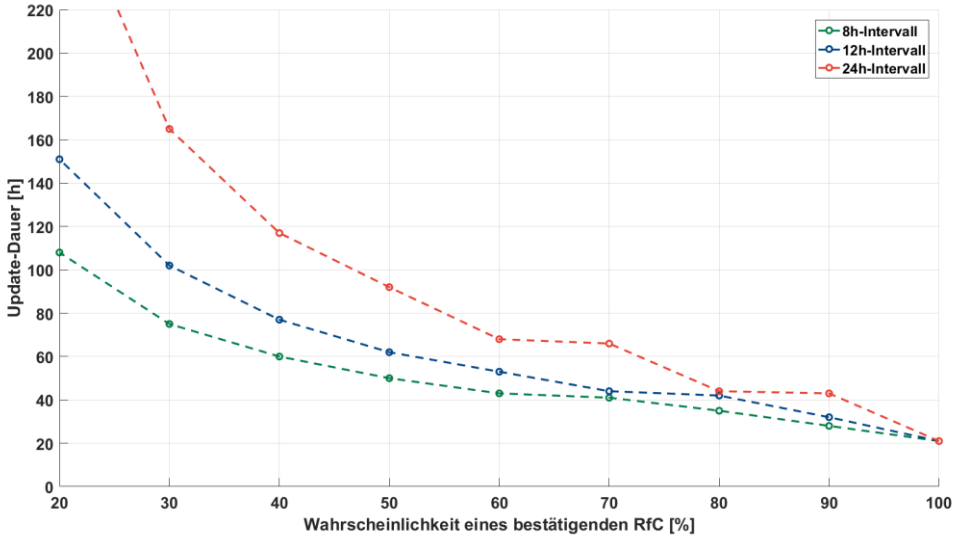


Abbildung 23: Einfluss der Wahrscheinlichkeit, dass ein Update angenommen wird

Abbildung 23 zeigt den Einfluss, wenn die Wahrscheinlichkeit dafür variiert wird, dass das Update angenommen wird. Dieser ist im Bereich über 50 % nahezu linear und anschließend progressiv. Außerdem skaliert die Update-Dauer um einen Faktor nach oben, wenn das Intervall der Nachfrage nach der Bestätigung des Updates variiert wird.

Neben des bereits diskutierten negativen Einfluss einer zu häufigen Nachfrage auf die Wahrscheinlichkeit der Annahme, kann diese Wahrscheinlichkeit jedoch wieder über die Gestaltung der Nachricht an den Kunden positiv beeinflusst werden. Je besser die Kunden über die Inhalte, mögliche Konsequenzen, die Installationszeit und weitere Faktoren informiert werden, umso besser können sie die tatsächliche Relevanz des Updates für ihr Fahrzeug beurteilen und sich entsprechend einrichten, das Update vorzunehmen.

Für den weiteren Verlauf und der Dringlichkeit des Updates wird eine Update-Intervall von 12 Stunden eingeplant. Im Vergleich zum 8 Stunden Intervall wird hierdurch eine höhere Wahrscheinlichkeit von 60 % für die Annahme angenommen. Die Update-Dauer sinkt auf rund 52,9 Stunden.

Wahrscheinlichkeit eines Download-Erfolges

Auf das 90%-Quantil der Update-Dauer hat die Wahrscheinlichkeit eines Download-Erfolges keinen signifikanten Einfluss. Dieses bleibt weitestgehend konstant. Erst wenn zu viele Fahrzeuge

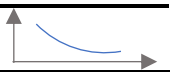
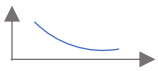
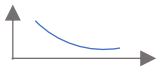
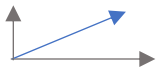
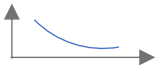
in diesem Prozessschritt auf Grund des zu häufigen Fehlschlagens des Downloadversuches ausfallen, wird das 90%-Quantil negativ beeinflusst. Dies müssten jedoch mehr als 10 % der Fahrzeugflotte betreffen, um einen Einfluss zu haben. Es wird angenommen, dass dieser Wert – allein für diesen Prozessschritt – unwahrscheinlich ist. Die Update-Dauer bleibt damit weiter bei rund 52,9 Stunden.

Anzahl an Installationsversuchen pro Tag

Nach dem erfolgreichen Download es Updatepaketes wird die Installation angestoßen. Hier wurde ein Parameter hinterlegt, um Anzahl der Zeitpunkte zu beschreiben, die versucht wird das Update zu installieren. Dies wurde initial auf 1 gesetzt, das heißt, dass alle 24 Stunden versucht wird die Installation zu starten. Aufgrund der Dringlichkeit des Updates wird dieser Wert jedoch auf 3 erhöht. Dies hieße, dass dem Fahrer alle 8 Stunden der Versuch eine Installation vorgeschlagen wird. Hierdurch kann die Update-Dauer auf 48,4 Stunden reduziert werden. Zu berücksichtigen ist jedoch, dass durch das häufigere Vorschlagen des Updates die Wahrscheinlichkeit der Update-Annahme sinken kann, da der Nutzer zu häufig „gestört“ wird und einen Automatismus für das Ablehnen entwickelt.

4.3.3 Schlussfolgerung

Die Analyse zeigt den unterschiedlichen Einfluss der verschiedenen Parameter. Bei der Auswahl der Kenngröße für verschiedene Parameter zeigt sich, dass ein Kompromiss zwischen der Geschwindigkeit des Update-Prozesses und weiteren wichtigen Faktoren, wie der (wirtschaftlichen) Effizienz, aber auch der Akzeptanz des Kunden etc. eingegangen werden muss. Diese wurden im Optimierungsprozess exemplarisch diskutiert. Tabelle 2 fasst die vorgenommenen Optimierungen zusammen, protokolliert die veränderte Update-Dauer und zeigt, wie sich die Variierung einzelner Parameter auf die Update-Dauer auswirkt.[12]

Prozess-Parameter	Optimierung	Update-Dauer	Einflussart
Ausgangsdauer		215,9 Std.	
Anzahl Server-Slots / Anzahl Fahrzeuge	Steigerung des relativen Verhältnisses zueinander von 1% auf 2,5%	211,5 Std.	
Verteilung der Übertragungstechnik	Folgt aus dem Szenario: WiFi 20%, 3G 30%, 4G 50%.	211,5 Std.	
Größe des Updatepaketes	Folgt aus dem Szenario: 10 MB	211,5 Std.	
Erreichbarkeit des Fahrzeugs	Wird bei Shoulder-Tap Methode auf 90% festgelegt	212,2 Std.	

[12] Auf der Ordinate des symbolischen Diagramms in Tabelle 2 ist die Update-Dauer und auf der Abszisse sind die jeweiligen variierten Parameter aufgetragen.

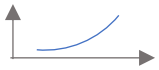
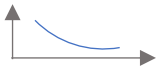
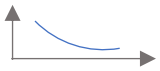
Wahrscheinlichkeit der richtigen Konfiguration	Korrekturfaktor, wird auf 99,5% gesetzt	212,2 Std.	
Zeitraum der wiederholten Erinnerung des Updates	Der Nutzer wird nun statt alle 24 Std. alle 12 Std. an das Update zur Autorisierung erinnert.	86,8 Std.	
Wahrscheinlichkeit für die Annahme des Updates	Durch die Security-Relevanz des Updates wird die Wahrscheinlichkeit von 25% auf 60% nach oben korrigiert.	52,9 Std.	
Anzahl an Installationsversuchen pro Tag	Aufgrund der Dringlichkeit wird die Anzahl von 1 auf 3 erhöht.	48,4 Std.	

Tabelle 2: Zusammenfassung der Prozessoptimierung

5 PROJEKTABSCHLUSS

In diesem Projekt wurden die Ergebnisse des ersten Projektes weiter vertieft und gegengeprüft. Zunächst wurden die vorgeschlagenen Klassifizierungen für Softwareupdates miteinander verglichen. Dabei wurde festgestellt, dass eine Kombination und Ergänzung dieser Kombinationen miteinander aufgrund der zu unterschiedlichen Betrachtungswinkel nur in einer neuen Klassifizierung mit zu vielen Klassen resultieren würde. Die Vorteile einer Klassifizierung würden sich durch eine zu sperrige Anwendbarkeit nivellieren. Vielmehr ist eine Einstufung von Updates mittels Attribute zu empfehlen – siehe Abschnitt 2.5.

Anschließend finden sich in Kapitel 3 eine deutlich überarbeitete Version des im ersten Projekt zu Softwareupdates Over-the-Air entworfenen idealisierten Prozesses. Dieser wurde in allen Aktivitäten in der inhaltlichen Tiefe überarbeitet – z. B. um die Konsequenzen der verschiedenen und möglichen Attribute – , aber auch an vielen Stellen korrigiert, so dass Aktivitäten entfallen sind und andere hinzukamen. Ein besonderer Untersuchungsaspekt waren die Anforderungen der UN ECE ‚Taskforce on Cyber Security and OTA Issues' die nun zu den einzelnen Aktivitäten des idealisierten Prozesses verknüpft sind.

Ein drittes Arbeitspaket des Projektes war die Simulation dieses Prozesses, um verschiedene Use Case miteinander zu vergleichen und den Einfluss der verschiedenen Prozessparameter zu untersuchen. Die Simulation wurde dafür im AQI in Matlab programmiert. Allein durch die Modellierung und anschließende Implementierung fand bereits eine Validierung der Prozesskette statt, die entscheidend zur Detaillierung der Prozessbeschreibung beigetragen hat.

Kern dieses Projekt ist diese Prozessexplikation, die einen ganzheitlichen, idealisierten OTA-Prozess von der Generierung des Updates über die Verteilung hin zur Installation beschreibt. Dies kann auf Verbandsebene als Diskussionsgrundlage genutzt werden, intern einem Abgleich gegenüber den Anforderungen der UN ECE oder der Strukturierung des eigenen OTA-Prozesses dienen.

Anhang

Entfallene Aktivitäten gegenüber dem Projekt OTA-1

OTA-G-010 [entfällt]

~~Input: Information über eine Fehlfunktion im Fahrzeug. Die Quelle für die Information kann unterschiedlich sein: sie kann sowohl intern durch eigene Beobachtungen oder extern durch Kommunikation mit der Regulatorischen Behörde, den Kunden, der Presse o. ä. erfolgen.~~
~~Der ***Update-Koordinator*** analysiert die Informationen und identifiziert den Bedarf eines Software-Updates für ein betroffenes Bauteil / ein betroffenes System.~~
~~Output: Updatebedarfsmeldung inkl. Fehlerbild und mögliche Ursache zur weiteren Nachverfolgung~~

Für die Identifizierung eines Update-Bedarfs sind entsprechende Prozesse bereits etabliert. Daher entfällt diese Aktivität im OTA-Prozess, da außerdem hierzu keine OTA-spezifischen Anforderungen existieren. Der OTA-Prozess startet nun direkt mit einem bereits identifizierten Updatebedarf bzw. den definierten Anforderungen / dem Lastenheft des Updates.

Dennoch sei darauf hingewiesen, dass der Bedarf nach einem Softwareupdate grundlegend über drei verschiedene Kanäle erfolgen kann:

(1) Fehlerabstellprozess

(2) Strategische Produktentwicklung

(3) Kunden-Feedback oder Kundenwunsch, z. B. zum Freischalten einer Funktion im Fahrzeug [13]

OTA-G-040 [entfällt]

~~Der ***Update-Koordinator*** kommuniziert den Updatebedarf an den *Entwicklungsverantwortlichen*.~~
~~Output: dokumentierte Updatebedarfsmeldung~~

[13] Die Vernetzung der Fahrzeuge ermöglicht die direkte Kommunikation mit dem Kunden. Unter den Nutzern gibt es Gruppen, die gerne Feedback an Hersteller zurückmelden und sich aktiv an Produktentwicklungen beteiligen.

Diese Aktivität passte nur, wenn der Prozess nur Updates für Fehlerbehebungen berücksichtigt. Da nun jedoch weitere Use Cases berücksichtigt werden, entfällt diese Aktivität. Bzw. sie geht in die neue Aktivität OTA-G-051 auf, die nun den Prozess aus der relevanten Abteilung heraus startet.

~~OTA-G-050 [entfällt]~~

~~Der ***Update-Koordinator*** definiert das Lastenheft für das Software-Update.~~
~~Output: Lastenheft für Software-Update~~

Begründung siehe OTA-G-040.

~~OTA-G-080 [entfällt]~~

~~Input: Fahrzeugkonfiguration / Fahrzeugdaten~~
~~Der ***Entwicklungsverantwortliche*** analysiert die übermittelte Fahrzeugkonfiguration / die extrahierten Fahrzeugdaten und unterstützt so ggf. den *Entwickler* mit den Ergebnissen der Analyse der übermittelten System- / Diagnosedaten zum Fahrzeugzustand.~~
~~Output: Diagnoseergebnisse~~[14]

Entfällt, da interne Prozesse abgebildet werden, für die keine Vorgaben benötigt werden. Die Inhalte der Aktivität sind durch die Aktivität OTA-G-090 abgedeckt.

~~OTA-G-100 [entfällt]~~

~~Der ***Entwickler*** übermittelt das Software-Update an den Update-Server des OEM und informiert den *Entwicklungsverantwortlichen* über den neuen Softwarestand.~~
~~Output: dem OEM verfügbares Software-Update; Mitteilung über Verfügbarkeit des Updates~~

Entfällt, da interne Prozesse abgebildet werden, für die keine Vorgaben benötigt werden. Die Inhalte der Aktivität sind durch den Kommunikationsfluss zwischen Aktivität OTA-G-090 und OTA-G-110 abgedeckt.

[14] Die Diagnoseergebnisse können verschieden ausgeprägt sein. Basis sind Marktdaten betroffener Fahrzeuge, möglich sind auch die Fahrzeugkonfigurationen unter denen die Fehler auftreten, bis hin zu Benutzeraktivitäten bei denen die Fehler aufgetreten sind.

OTA-G-140 [entfällt]

~~Der ***Update-Koordinator*** zeichnet das Software-Update für die Umsetzung des Rückrufs an den Kundenfahrzeugen frei.~~
~~Output: freigezeichnetes Update~~

Die Aktivität entfällt und ist durch die Aktivität OTA-G-130 abgebildet, die den internen Freigabeprozess des jeweiligen Unternehmens widerspiegelt. Eine Differenzierung in zwei separate Aktivitäten erzielt keinen Mehrwert, zumal die Aktivität des Freigebens nur bedingt OTA-spezifisch ist. Nichtsdestotrotz kann sie einen erheblichen Einfluss auf die Performance der Prozesskette haben.

OTA-V-070 [entfällt]

~~Der ***Digitale Dienstleister*** aktualisiert die Fahrzeug-Datenbank entsprechend der Rückmeldung des Fahrzeuges.~~
~~Output: Nachricht über aktualisierte Fahrzeugdatenbank~~
~~Anm.: Wenn die Fahrzeugkonfiguration nicht kompatibel ist, das heißt, nicht den Anforderungen des Updates entspricht, entfällt das Fahrzeug aus dem weiteren Updateprozess. Dieser Umstand sowie dessen Ursache dafür werden vom Digitalen Dienstleiser dokumentiert und im Rahmen eines Update-Reports an den Auftraggeber übermittelt.~~

Diese Aktivitäten kann entfallen, da sie automatisiert durch OTA-V-060 stattfinden kann und keines zusätzlichen administrativen Aufwandes benötigt.

OTA-V-010 [entfällt]

~~Input: freigezeichnetes Update-Paket~~
~~Der ***Update-Koordinator*** leitet das freigezeichnete Update an den *Digitalen Dienstleister* weiter.~~
~~Output: freigezeichnetes Update-Paket~~

Entfällt, da die Aktivität in OTA-G-141 (das Updatepaket wird an den *Digitalen Dienstleister* gesandt) und OTA-V-020 (der *Digitale Dienstleister* empfängt das Updatepaket) aufgeht.

OTA-V-110 [entfällt]

~~Das ***Fahrzeug*** verwaltet den Genehmigungsprozess und leitet die Entscheidung über die Installation des Updates weiter.~~
~~Output: Ergebnis des Genehmigungsprozesses~~
~~**Konditionen**: **(A)** bei erteilter Genehmigung wird der Update-Koordinator entsprechend benachrichtigt. **(B)** bei nicht erfolgter Genehmigung kann das Update erneut angezeigt werden. Wird das Update endgültig abgelehnt, ist auch dies dem Update-Koordinator mitzuteilen.~~

Diese Aktivität verläuft automatisch und bedarf keiner gesonderten Explikation, da hierdurch keine besonderen Anforderungen resultieren. Weitere Anforderungen sind in die Aktivität OTA-V-120 übernommen.

OTA-V-130 [entfällt]

~~Der ***Digitale Dienstleister*** initiiert den Start des Downloads (z.B. durch Versenden einer Download-Referenz).~~
~~Output: Download-Referenz~~
~~Anm.: A) diese Aktivität kann auch schon direkt nach Freischaltung des Updates passieren, d.h. vor der eigentlichen Autorisierung des Updates durch den Halter. Zur Steigerung der Kundenzufriedenheit kann das Update bereits geladen werden, während die Bestätigung noch aussteht. Eine Installation kann somit verzögerungsfrei direkt nach der Bestätigung erfolgen. Dies ist jedoch mit höheren Kosten verbunden, z.B. bei einer dann nicht folgenden Autorisierung des Updates. B) Ein Update kann auch in mehreren ‚Wellen' an die Fahrzeuge verteilt werden, um die Zugriffe auf die Serverstruktur zeitlich zu verteilen.~~

Die Aktivität kann entfallen, da sie bereits in der Benachrichtigung des Fahrzeugs (OTA-V-050) impliziert wird.