

TEXTE

13/2023

Abschlussbericht

Analyse der softwarebasierten Einflussnahme auf eine verkürzte Nutzungsdauer von Produkten

von:

Melanie Jaeger-Erben
Institut für Sozialinnovation Consulting UG, Berlin

Erik Poppe, Eduard Wagner, Anton Schaefer
Fachgebiet Transdisziplinäre Nachhaltigkeitsforschung in der Elektronik, Technische Universität
Berlin

Jan Druschke
Fraunhofer Institut für Zuverlässigkeit und Mikrointegration, Berlin

Jens Gröger, Felix Behrens
Öko-Institut e.V., Berlin

Herausgeber:

Umweltbundesamt

TEXTE 13/2023

Ressortforschungsplan des Bundesministeriums für Umwelt,
Naturschutz, nukleare Sicherheit und Verbraucherschutz

Forschungskennzahl 3719 37 309 0
FB000891

Abschlussbericht

Analyse der softwarebasierten Einflussnahme auf eine verkürzte Nutzungsdauer von Produkten

von

Melanie Jaeger-Erben

Institut für Sozialinnovation Consulting UG, Berlin

Erik Poppe, Eduard Wagner, Anton Schaefer

Fachgebiet Transdisziplinäre Nachhaltigkeitsforschung in
der Elektronik, Technische Universität Berlin

Jan Druschke

Fraunhofer Institut für Zuverlässigkeit und
Mikrointegration, Berlin

Jens Gröger, Felix Behrens

Öko-Institut e.V., Berlin

Im Auftrag des Umweltbundesamtes

Impressum

Herausgeber

Umweltbundesamt
Wörlitzer Platz 1
06844 Dessau-Roßlau
Tel: +49 340-2103-0
Fax: +49 340-2103-2285
buergerservice@uba.de
Internet: www.umweltbundesamt.de

[f/umweltbundesamt.de](https://www.facebook.com/umweltbundesamt.de)

[t/umweltbundesamt](https://twitter.com/umweltbundesamt)

Durchführung der Studie:

Institut für Sozialinnovation Consulting (ISIconsult)
Köpenicker Straße 325
12555 Berlin

Abschlussdatum:

Oktober 2021

Redaktion:

Fachgebiet Beratungsstelle nachhaltige Informations- und Kommunikationstechnik
(Green-IT)
Marina Köhn

Publikationen als pdf:

<http://www.umweltbundesamt.de/publikationen>

ISSN 1862-4804

Dessau-Roßlau, Januar 2023

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autorinnen und Autoren.

Kurzbeschreibung: Analyse der softwarebasierten Einflussnahme auf eine verkürzte Nutzungsdauer von Produkten

Seit einigen Jahren steigt die Anzahl von smarten Elektrogeräten und vernetzten Systemen im Alltag und den Haushalten rasant an. Damit nimmt auch das Risiko von Softwareobsoleszenz zu, das heißt, der Verkürzung der Nutzungs- und Lebensdauer eines technisch funktionalen Geräts aufgrund von Software. Verkürzte Produktlebensdauern stellen wiederum ein Umweltproblem dar, daher ist umweltpolitisches Handeln geboten. Das Projekt „Analyse der softwarebasierten Einflussnahme auf eine verkürzte Nutzungsdauer von Produkten“ hat sich im Auftrag des Umweltbundesamts mit dem Phänomen Softwareobsoleszenz, seinen Ursachen, der aktuellen Verbreitung sowie der Marktentwicklung von vernetzten Geräten befasst. Mit dem Fokus auf Obsoleszenzrisiken im Bereich Kompatibilität, Funktionalität und Sicherheit wurden sowohl empirische Fälle von Softwareobsoleszenz als auch exemplarische Fälle unterschiedlicher Wirkungszusammenhänge von Software und Hardware untersucht. Es zeigt sich, dass Softwareobsoleszenz öffentlich bereits wahrgenommen wird, aber noch nicht weit verbreitet ist. Dennoch zeigen Trendanalysen, dass vor allem SmartHome-Systeme und damit komplexe Obsoleszenzrisiken in den nächsten Jahren zunehmen werden. Die größte Herausforderung besteht jedoch nicht in technisch-physikalischer Hinsicht, sondern vor allem in den wirtschaftlichen und organisationalen Bedingungen, die zum frühzeitigen Ausfall der Software sowie ganzer Produktsysteme führen, wie das Einstellen von technischem Support oder die fehlende Kompatibilität verschiedener Systeme. Regulatorische Maßnahmen, wie die Definition von Mindestanforderungen an Software für den Marktzugang und die Verpflichtung zu Transparenz auf Seiten der Anbieter, sind somit genauso nötige wie wirkungsvolle politische Instrumente. Gleichzeitig besteht weiterhin Forschungsbedarf, vor allem im Hinblick auf technische, wirtschaftliche und rechtliche Folgenabschätzungen.

Abstract: Analysis of software-induced shortening of the useful life of products

For several years now, the number of smart electrical devices and networked systems in everyday life and households has been increasing rapidly. This also increases the risk of software obsolescence, i.e., the shortening of the useful life of a technically functional device due to software. Shortened product lifetimes in turn pose an environmental problem, so environmental policy action is required. The project "Analysis of software-induced shortening of the useful life of products" was commissioned by the Federal Environment Agency and dealt with the phenomenon of software obsolescence, its causes, the current prevalence as well as the market development of networked devices. With a focus on obsolescence risks in the areas of compatibility, functionality and security, empirical cases of software obsolescence as well as exemplary cases of different interdependencies of software and hardware were investigated. It was revealed that software obsolescence is already publicly perceived but is not yet widespread. Nevertheless, trend analyses show that smart home systems in particular, and thus complex obsolescence risks, will increase in the coming years. The greatest challenge, however, is not in technical-physical terms, but primarily in the economic and organisational conditions that lead to the premature failure of software as well as entire product systems, such as the discontinuation of technical support or the lack of compatibility between different systems. Regulatory measures, such as the definition of minimum requirements for software for market access and the obligation for transparency on the part of the providers, are thus both necessary and effective political instruments. At the same time, there is still a need for research, especially with regard to technical, economic and legal impact assessments.

Inhaltsverzeichnis

Inhaltsverzeichnis.....	6
Abbildungsverzeichnis.....	8
Tabellenverzeichnis.....	9
Abkürzungsverzeichnis.....	10
Zusammenfassung.....	12
Summary.....	21
1 Einleitung.....	29
1.1 Hintergrund und Relevanz.....	29
1.2 Fragestellung und Vorgehen.....	29
1.3 Gegenstandsbeschreibung: Was ist Softwareobsoleszenz?.....	31
2 Bestandsaufnahme und Stand der Forschung.....	36
2.1 Hintergrund: Entwicklung der IKT-Industrie.....	37
2.2 Forschung zum Thema softwarebedingte Obsoleszenz.....	42
2.3 Zusammenfassung und Fazit.....	49
3 Wirkungszusammenhänge von Soft- und Hardware.....	50
3.1 Software Schichtenmodell.....	50
3.2 Übertragungsstandards in lokalen Netzwerken.....	52
3.3 Erweitertes Schichtenmodell und Abhängigkeitskategorien.....	53
3.4 Beispiel hoher Hardwareabhängigkeit: Smartphone App „Corona-Warn-App“.....	55
3.5 Beispiel starker Verschränkung von Hard- und Software: „Smarte Leuchtmittel mit Steuerungszentrale“ Philips Hue.....	60
3.6 Beispiel hoher Softwareabhängigkeit: Smartes Heizkörperthermostat.....	64
3.7 Zusammenfassung der Wirkungszusammenhänge von Soft- und Hardware.....	67
4 Untersuchung der Marktentwicklung.....	69
4.1 Fragestellungen und Vorgehensweise.....	69
4.2 Haushaltsentwicklung in Deutschland.....	71
4.3 Hochrechnung der Geräteanzahl in smarten Haushalten.....	72
4.4 Marktuntersuchungen ausgewählter Produkte.....	74
4.5 Zusammenfassung und Fazit.....	84
5 Analyse empirischer Fälle von softwarebedingter Obsoleszenz.....	86
5.1 Kernaspekte und Ziele der Untersuchungen.....	87
5.2 Zusammenfassung und Fazit der Fallstudien.....	95
6 Wahrnehmungen und Erfahrungen von Konsument*innen.....	99

6.1	Fragestellungen und Vorgehen	99
6.2	Ergebnisse der Auswertung von Online-Foren und Produktbewertungen	102
6.3	Ergebnisse der repräsentativen Befragung	104
6.4	Fazit.....	109
7	Wahrnehmungen und Erfahrungen von Expert*innen und Praktiker*innen	111
7.1	Fragestellungen und Vorgehensweise	111
7.2	Ergebnisse der Begleitkreis-Workshops	111
7.3	Ergebnisse der Interviews	113
7.4	Zusammenfassung und Fazit	117
8	Ableitung produktpolitischer Maßnahmen	119
8.1	Zielsetzung	119
8.2	Regulatorische Ansatzpunkte.....	119
8.3	Umweltpolitische Handlungsempfehlungen	128
8.4	Steckbriefe	134
8.5	Zusammenfassung	134
9	Weitere Forschungsbedarfe	137
10	Quellenverzeichnis	139
A	Leitfaden Akteursinterviews.....	148
B	Steckbriefe zu politischen Handlungsempfehlungen.....	151
B.1	Obsoleszenz durch externe Abhängigkeiten	151
B.2	Obsoleszenz bei Systemprodukten	155
B.3	Direkte softwarebedingte Obsoleszenz.....	159
C	Anhang: Fragebogen für repräsentative Nutzer*innen-Umfrage.....	164

Abbildungsverzeichnis

Abbildung 1: LEUF-Schema.....	34
Abbildung 2: Auswahl möglicher Obsoleszenzfaktoren entlang des OSI-Schichtenmodells.....	39
Abbildung 3: Durchschnittliche Mitwirkung und Abhängigkeiten pro Paketsprache auf Github.com im Jahr 2019.....	40
Abbildung 4: Ergebnisse der Metasuche in IEEEExplore zu Software Obsoleszenz.....	42
Abbildung 5: Relevante Veröffentlichungen pro Jahr im IEEEExplore.....	43
Abbildung 6: Qualitätskriterien der ISO/IEC 25010.....	47
Abbildung 7: Zusammenspiel von Soft- und Hardware in modernen Elektrogeräten.....	48
Abbildung 8: Wirkungsgefüge innerhalb einer Schicht.....	54
Abbildung 9: Schichtenmodell erweitert um Abhängigkeitsverhältnisse.....	54
Abbildung 10: Vorgehensmodell für die Marktanalyse im Bereich SmartHome.....	70
Abbildung 11: Entwicklung der Haushalte in Deutschland 2019 und 2030.....	71
Abbildung 12: Entwicklung der smarten Haushalte in Deutschland bis 2030.....	72
Abbildung 13: Anwendungen in smarten Haushalten in Deutschland.....	73
Abbildung 14: Anwendungen in allen Haushalten in Deutschland.....	74
Abbildung 15: SmartHome-Ökosystem.....	76
Abbildung 16: SmartTVs und deren Betriebssysteme 2008 bis 2020.....	77
Abbildung 17: Marktanteile Smartphone-Hersteller in Europa von 2015 bis 2020.....	80
Abbildung 18: Bewertungsmatrix von SmartHome-Anbietern.....	82
Abbildung 19: Beziehungsdiagramm SmartHome-Anbieter.....	84
Abbildung 20: Das Systemdiagramm in dem Dimensionen Milieu (horizontal) und Material (vertikal) aufgespannt; exemplarische Darstellung der Charakteristika in roten Kästen.....	88
Abbildung 22: Verzug bei Sicherheitsupdates bei Betriebssystemen.....	92
Abbildung 23: Auszüge aus der End of Life Ankündigung von Belkin vom 28.04.2020.....	94
Abbildung 24: Heatmap der zusammengefassten Ursachen (oberer Teil) und Wirkungen (unterer Teil) der untersuchten Fallstudien.....	97
Abbildung 25: Erfahrungen von Nutzenden aus Recherche in Foren 1.....	101
Abbildung 26: Erfahrungen von Nutzenden aus Recherche in Foren 2.....	101
Abbildung 27: Smarte Geräte im Haushalt [%] (ausgenommen Smartphone, Notebook, Tablet).....	104
Abbildung 28: Wahrgenommene Verschlechterungen beim Smartphone im Verhältnis zur Nutzungsdauer [%].....	105
Abbildung 29: Wahrgenommene Verschlechterungen beim Smartphone im Verhältnis zur Altersgruppe der Nutzer*innen.....	106
Abbildung 30: Die am häufigsten genannten Anschaffungswünsche, Angaben in % (das n in Klammern gibt an, wie viele Befragte das Gerät noch nicht besitzen und daher die Frage gestellt bekamen).....	107
Abbildung 31: Politikinstrumente zur nachhaltigen Markttransformation.....	128

Tabellenverzeichnis

Tabelle 1: Auswahl verwandter Konzepte und Begriffe	32
Tabelle 2: Beispiele zur Differenzierung von direkter und indirekter Softwareobsoleszenz	33
Tabelle 3: Fokussierte Medien in der Literaturrecherche	36
Tabelle 4: Ursachen von Softwareobsoleszenz (n. Sandborn 2007)	44
Tabelle 5: Übersicht der OSI Schichten.....	50
Tabelle 6: Obsoleszenzrisiken Beispiel 1: Smartphone Corona-Warn-App.....	60
Tabelle 7: Obsoleszenzrisiken Beispiel 2: Smarte Leuchtmittel	63
Tabelle 8: Obsoleszenzrisiken Beispiel 3a: Smartes Thermostat Eve.....	66
Tabelle 9: Obsoleszenzrisiken Beispiel 3b: Smartes Thermostat Tado	67
Tabelle 10: mögliche Kategorisierungen für SmartHome-Geräte.....	75
Tabelle 11: Eigenschaften von Smart-TV-Betriebssystemen	78
Tabelle 12: Indikatoren für das Risiko softwarebedingter Obsoleszenz von SmartHome-Systemen	81
Tabelle 13: Ausgewählte Fallbeispiele zur softwarebedingten Obsoleszenz bei Konsumgütern	86
Tabelle 14: Schwerpunkte der Vor-Recherche zur Entwicklung des Fragebogens	100
Tabelle 15: Auswertung Nutzer*innen-Erfahrungen aus Online-Foren und Produktbewertungen.....	102
Tabelle 16: Im Interview vorgeschlagene Maßnahmen.....	117
Tabelle 17: Übersicht zu wesentlichen politischen Entwicklungen und Forderungen mit Bezug zu Softwareobsoleszenz (nach Erscheinungsjahr geordnet)	120
Tabelle 18: Anforderungen an software-relevante Aspekte in den Ökodesign- Richtlinien (RL (EU) 2009/125/EG)	125

Abkürzungsverzeichnis

AP	Arbeitspaket
API	Application Programming Interface
BGB	Bürgerliches Gesetzbuch
Bits	Informationseinheit
BMU	Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit
BMWi	Bundesministerium für Wirtschaft und Energie
BSI	British Standards Institution
BSI	Bundesamt für Sicherheit und Informationstechnik
COG	Component Obsolescence Group
CPS	Cyber-Physical-System
EHS	European Home System
EIBA	European Installation Bus Association
EIBA	Europäischer Installationsbus
ELV	End of Life Vehicle Directive
EMI	Electromagnetic inference / Elektromagnetische Störung
ENS	Exposure Notification System
GSMA	Global System for Mobile Communications Association
HTTPS	Hypertext Transfer Protocol Secure
HW	Hardware
IKT	Informations- und Kommunikationstechnologie
IoT	Internet of Things
IP	Internetprotokoll
IT	Informationstechnologie
LEUF	Legal, Executable, Usable, Functional
MFI	ursprünglich "Made for Ipad"
NGO	Non-Governmental Organisation
OSI	Open Systems Interconnection
OTA-Updates	over-the-air-Updates
PPCP	Privacy Preserving Contact Tracing Protocol
REACH	Registration, Evaluation, Authorisation and Restriction of Chemicals
RED	Radio Equipment Directive
REST	Representational State Transfer
RKI	Robert-Koch-Institut
RoHS	Restriction of Hazardous Substances in electrical and electronic Equipment
RPID	Rolling Proximity Identifier
SAR	Software aging and rejuvenation
SDK	Software Development Kit

SobO	Software-bedingte Obsoleszenz. Kurztitel für „Analyse der softwarebasierten Einflussnahme auf eine verkürzte Nutzungsdauer von Produkten“
SPCN	Software Product Change Notification
SW	Software
UBA	Umweltbundesamt
UPnP	Universal Plug and Play
UX	Usability and Userexperience
VZBV	Verbraucherzentrale Bundesverband
VzWv	Vorbereitung zur Wiederverwendung

Zusammenfassung

Der Anteil der alltäglich genutzten Geräte, die von Software abhängig sind, nimmt stetig zu. Nicht nur bei klassischen digitalen Geräten wie Smartphones und Laptops, auch im Bereich der Unterhaltungselektronik steigt der Anteil an Geräten, deren Funktionalität durch integrierte Mikrocontroller und eine zugehörige Software ermöglicht wird. Ein Trend, der sich zunehmend auch auf Haushaltsgeräte, Anlagen der Gebäudetechnik sowie Fahrzeuge ausweitet. Immer häufiger entscheidet daher auch die Qualität der Software über die Nutzungsdauer, Funktionalität und Zuverlässigkeit von Geräten. Über die Software können zum einen die Nutzungsdauer und der Nutzungskomfort von Geräten definiert oder verändert werden. Zum anderen bleiben die Sicherheit der Nutzung und der Fortbestand der Kompatibilität zwischen Hardware und Software sowie zwischen miteinander vernetzten Geräten eine konstante Herausforderung. Bei langlebigen Produkten kann softwarebedingte Obsoleszenz zu einem bedeutenden Problem für die Haltbarkeit und Zuverlässigkeit der Geräte werden.

Aus der Umweltperspektive ist das Problem offensichtlich und wird vereinzelt bereits durch aktuelle Strategien in der nationalen und europäischen Gesetzgebung adressiert, wie der Umweltpolitischen Digitalagenda (BMU 2020) oder dem europäischen Aktionsplan zur Kreislaufwirtschaft (EC 2020). Im Bereich der sozio-ökologischen Forschung ist das Phänomen der Softwareobsoleszenz bisher wenig erforscht und verstanden. Die öffentliche Aufmerksamkeit für das Thema sowie das wachsende Problemverständnis durch verschiedene gesellschaftliche Interessengruppen zeigt jedoch den wachsenden Bedarf nach Systematisierung und Erforschung der Ursachen, Phänomene und Problemlösungsansätze im Zusammenhang mit softwarebedingter Obsoleszenz.

Vor diesem Hintergrund hat das Umweltbundesamt Ende 2019 das Projekt „Analyse der softwarebasierten Einflussnahme auf eine verkürzte Nutzungsdauer von Produkten“ (kurz: „SobO“; Förderkennzeichen: UFOPLAN 3719 37 309 0) ins Leben gerufen. Das Projekt hatte zum Ziel, systematisch die Erscheinungsformen und Ursachen softwarebedingt verkürzter Nutzungs- und Lebensdauern zu untersuchen und auf dieser Basis politische Handlungsoptionen abzuleiten.

Um fundierte Vorschläge für die Ursachenbekämpfung aufzuzeigen, wurden unter Einbezug inter- und transdisziplinärer Perspektiven und mit einer breiten sozial-, ingenieur-, wirtschafts- und politikwissenschaftlicher Expertise verschiedene Schlüsselsektoren (Konsum und Verbraucherschutz, Produktion und Produkttest, Hardware- und Software-Entwicklung, politische Rahmenbedingungen) in ihrem Zusammenspiel betrachtet. Methodisch wurden dabei neben qualitativen und quantitativen Befragungen verschiedene empirische Fallstudien, eine Markt- und Trendanalyse sowie vertiefende Ursache-Wirkungs-Analysen von vernetzten Gerätesystemen angewendet. Im Folgenden werden die zentralen Ergebnisse entlang der Kapitelstruktur zusammengefasst.

Definition softwarebedingte Obsoleszenz: Dynamik des Produktsystems beachten!

Es fehlt bisher an einer formalen Definition von Softwareobsoleszenz, und das Phänomen ist derzeit nur wenig wissenschaftlich untersucht. In der vorhandenen Fachliteratur konnten bis heute zwar bereits verschiedene zentrale Wirkprinzipien identifiziert werden, eine generelle Definition zur besseren Ab- und Eingrenzung des Forschungsgegenstands und des Problemfelds fehlt bisher jedoch. Außerdem muss beachtet werden, dass Software keinem direkt beobachtbaren physischen Verschleiß unterliegt, wie er etwa bei greifbaren Gütern vorzufinden ist, deren Lebensdauer primär durch permanente thermische, oxidative und physische Belastungen begrenzt wird. Die Obsoleszenz von Software entsteht vielmehr innerhalb der

dynamischen Arrangements eines Produktsystems, in welchem Software spezifische Funktionen erfüllt oder nicht mehr erfüllen kann. Softwareobsoleszenz ist ein Sammelbegriff für Phänomene, bei denen die Qualität einer Software bis zu einem Grad abnimmt, bei der das mit ihr verbundene Produktsystem für eine signifikante Anzahl von Nutzenden nicht mehr hinreichend seine zuge dachte Funktion erfüllt. Softwareobsoleszenz wird ausgelöst durch die Software selbst (direkt) oder sich ändernden Anforderungen an die funktionalen und nicht-funktionalen Eigenschaften der Software (indirekt). Wenn mit Software verbundene Produktsysteme obsolet werden, kann die Software als möglicher Faktor eine mehr oder weniger starke Rolle spielen. Um die Funktionalität bzw. mögliche Risikofaktoren im Zusammenhang mit der Software eines Produktsystems besser einschätzen zu können, wurde im Projekt das sogenannte LEUF-Schema (siehe Kapitel 1.3.2) entwickelt. Entlang spezifischer Qualitäts- und Funktionsbedingungen (Legal, Executable, Usable, Functional) sollen Risikofaktoren für Softwareobsoleszenz in Anwendung und Nutzung besser formalisiert und in einer einfachen Ereigniskette abgebildet werden können.

Stand der Forschung: Bestimmung der Softwarequalität

Softwareobsoleszenz ist kein neues Phänomen und findet sich entlang der gesamten Computer- und Softwarehistorie, die ihren Ausgangspunkt in den 1940er Jahren mit der ersten Entwicklung nicht-ziviler Großrechner (Computer) nahm. Die Recherche hat offengelegt, dass das Thema der softwarebedingten Obsoleszenz innerhalb der Fachwelt in unterschiedlichen Diskursen bereits adressiert wird, das Problem an sich jedoch nur wenig wissenschaftlich untersucht und formalisiert ist. Im Allgemeinen werden die verschiedenen Erscheinungsformen oder Teilaspekte von softwarebedingter Obsoleszenz unter verschiedenen Begriffen wie dem „software rot“, „software erosion“, „software aging“ behandelt. Unabhängig vom Kontext und der Wirkungsweise teilen alle Konzepte dabei die Annahme, dass die langfristige Funktionsfähigkeit von Software von bestimmten Qualitätsmerkmalen abhängig gemacht werden kann, die sich jedoch im Zeitverlauf durch sich ändernde interne und externe Anforderungen ändern und degradieren können. Die IKT-Industrie ist seither geprägt von kurzen Marktzyklen, einem hohen Innovationstempo und einer zunehmenden Diversifizierung von Soft- und Hardwaretechnologien, sodass eine Verallgemeinerung, Generalisierung und Standardisierung relevanter Software-Qualitäten erschwert ist. Im Rahmen der Bestandsaufnahme und Recherche konnten zwar bereits zentrale Qualitätsstandards und Normen wie die ISO/IEC 25010 zur Bestimmung der Softwareproduktqualität identifiziert werden, deren tatsächliche Anwendung und Umsetzung in der Praxis bleiben aber fraglich und erfordern tiefergehende empirische Untersuchungen. Die Aufnahme von Mindestanforderungen und Kriterien für die Beschaffenheit von Software bei einzelnen Ökodesign-Richtlinien (RL (EU) 2009/125/EG) seit dem Jahr 2021 sowie die Ergänzung der Warenkaufrichtlinie (EU 2019/771) um Informationspflichten für digitale Elemente zeigen jedoch den wachsenden Bedarf für die stärkere wissenschaftliche Formalisierung und praktische Operationalisierung der nutzungs- und lebensdauerrelevanten Eigenschaften von Software.

Wirkungszusammenhänge von Soft- und Hardware: Eine Frage von Kompatibilität, Funktionalität und Sicherheit

Um die Wirkungszusammenhänge im Kontext softwarebedingter Obsoleszenz besser zu verstehen, wurde eine vergleichende Analyse verschiedener Produktsysteme in Anlehnung an das OSI-Schichtenmodells durchgeführt. Das OSI-Schichtenmodell (Open Systems Interconnection Model) ist ein weit verbreitetes Modell zur Beschreibung von Softwarearchitekturen. Für die Analyse wurden neben den funktionalen Abhängigkeiten innerhalb des OSI-Schichtenmodells auch Geschäftspraktiken und das Wirkungsgefüge von verschiedenen Akteuren (Soft- und Hardwareentwickler*innen, Vertriebsplattformen,

Anwender*innen) untersucht, aus denen sich in der Praxis Obsoleszenzrisiken ergeben. Für die langfristige Nutzbarkeit von softwarebasierten und -abhängigen Produkten wurden drei zentrale Wirkungskategorien ermittelt, die anhand von empirischen Fällen näher beleuchtet wurden. Daraus ergeben sich drei Stoßrichtungen, die softwarebedingte Obsoleszenz eindämmen:

- ▶ Langfristige Interoperabilität und Kompatibilität zwischen verschiedenen Geräten, Gerätegenerationen und Standards
- ▶ Bereitstellung und langfristige Gewährleistung von definierten Kernfunktionalitäten für Produkte innerhalb eines Produktsystems
- ▶ Wahrung der Sicherheit durch Konzepte wie dem Security by Design und qualitativ hochwertiger Software

Softwarebasierte und -abhängige Produkte sind in vernetzten Produktsystemen in der Regel permanenten Änderungen wie der Fehlerbeseitigung (bug fixes), oder Sicherheits- und Funktionsupdates unterworfen. Wobei die Analyse zeigt, dass die verschiedenen Akteure entlang der Softwarelieferkette über dominante Einflussmöglichkeiten verfügen, die sie zum Nachteil anderer Akteure einsetzen können (Schlüsselstellung von Gatekeepern). Hierzu zählt beispielsweise die Vorenthaltung von wichtigen Sicherheits- und Funktionsupdates oder Kompatibilität zu bestehenden Schnittstellen. Die Rolle des Gatekeepers liegt dabei nicht immer beim Hersteller oder Inverkehrbringer der Produkte, sondern auch bei Plattformbetreiber (Bsp. Vertragsbedingungen in Verkaufsportalen) und anderen dritten Akteuren in der Softwarelieferkette (Bsp. Bereitstellungsdauer von Firmware für Chipsets oder Betriebssysteme), deren indirekte Produktverantwortung stärker untersucht und regulatorisch adressiert werden sollte.

Markttrends und Marktentwicklung: SmartHomes auf dem Vormarsch

Für die bessere Einschätzung der künftigen Entwicklung und umweltpolitischen Relevanz wurde im Rahmen des Projekts eine Untersuchung der Marktentwicklung von Haushaltsgeräten bis zum Jahr 2030 vorgenommen. Im Fokus standen sogenannte SmartHome-Produktsysteme, wobei mit Smartphones und Smart TVs zwei besonders dynamische und aufgrund ihrer hohen Verbreitung relevante Produktsegmente vertiefend betrachtet wurden.

Die Untersuchung der Haushaltsentwicklung in Deutschland zeigt bis zum Jahr 2030 ein Marktpotential von 837.000 neuen Ein- und Zweipersonen-Haushalten für den SmartHome-Markt. Grundsätzlich lassen sich die Marktteilnehmer in Modulentwickler, Sensorhersteller und Gateway-Entwickler einteilen. In der Modulentwicklung am stärksten positioniert sind Unternehmen wie Phillips, OSRAM, SIEMENS oder Schneider Electric. Zudem nehmen Google, Amazon, Apple und Microsoft durch ihre Anwendungen im Gateway-Bereich eine dominante Rolle für die aktuellen und zukünftigen Marktentwicklungen ein. Für die betrachteten Produktgruppen lässt sich im Einzelnen festhalten, dass im Segment der SmartTVs die Betriebssysteme AndroidTV, Tizen und WebOS mit einem Anteil von 81,9% der zum Verkauf stehenden Geräte im deutschsprachigen Raum als wesentliche technologische Treiber gelten. Im Smartphone-Segment besetzen die vier Hersteller Samsung, Apple, Huawei und Xiaomi 84% des europäischen Marktes. Im Vergleich zum deutschen Smartphone-Markt, der bereits erste Anzeichen einer Sättigung aufweist, lag die Penetrationsrate auf dem europäischen Markt im Jahr 2018 bei 77,6% (Statista 2021).

Für die abschließende Einschätzung wird ein Vorschlag für ein Bewertungssystem mit Hilfe von sieben Indikatoren vorgestellt, welches eine grobe Abschätzung softwarebedingter

Obsoleszenzrisiken von Systemen und Plattformen ermöglicht. Das Bewertungssystem orientiert sich an den bereits im Vorfeld identifizierten Wirkungskategorien: Kompatibilität, Funktionalität und Sicherheit von Software.

Analyse empirischer Fälle von softwarebedingter Obsoleszenz: Das große Transparenzproblem

Im Bereich der Consumer-Elektronik (Haushaltselektronik, IKT, Unterhaltungselektronik) wurden für die vorliegende Studie verschiedene Fälle von softwarebedingter Obsoleszenz identifiziert und einer explorativen Analyse unterzogen. Es wurden insgesamt fünf empirische Fälle untersucht:

- ▶ Softwaregesteuerte Zählwerke bei Druckern
- ▶ Trade-up Programme bei Bluetooth-Lautsprechern unter Einsatz von softwaregesteuerten Abschaltvorrichtungen („kill switch“)
- ▶ Softwarebasierte Performance-Degradierung bei Smartphones
- ▶ Unterstützungszeitraum und Supportende bei Smartphone-Software
- ▶ End-of-Life von cloudbasierten Plattformen

Die Fallanalysen dienen in der Studie zur Validierung und Erweiterung der zu Grunde liegenden Wirkmechanismen und Entwicklung von praxisorientierten Handlungsempfehlungen.

In der Einzelfallbetrachtung fällt auf, dass Hersteller softwarebasierte Funktionsbeschränkungen und Konfigurationen überwiegend mit dem Hinweis auf qualitätssichernde Maßnahmen und Produktsicherheit rechtfertigen. Hierzu zählen exemplarisch Zählwerke und Verbrauchsmesser für Druckerpatronen, die ein optimales Druckerergebnis gewährleisten sollen, die softwarebasierte Serialisierung von Einzelkomponenten, um den Einbau von Ersatzteilen von Drittanbietern zu unterbinden oder die softwarebasierte Performance-Drosselung zur Verlängerung der Akkulaufzeit. In einem Fall wurde eine softwaregesteuerte Abschaltvorrichtung („kill switch“) dafür eingesetzt, mit Zustimmung der jeweiligen Kund*innen gebrauchte Produkte vollständig funktionsunfähig zu machen, um hierfür einen Gutschein für den Kauf eines Neuprodukts („Trade-In“) zu erhalten. Es lässt sich feststellen, dass die angemessene Anwendung von Schutzvorrichtungen durch Software in Geräten bisher eine rechtliche Grauzone darstellt und überwiegend im Ermessen des Herstellers liegt. In einigen Fällen kann dies jedoch den vorzeitigen Funktionsverlust von Hardware bewirken. Es gilt daher vertiefend zu klären, wie viel Schutz vor Übernutzung und Falschnutzung durch eine funktionsbegrenzende Intervention von Software überhaupt gerechtfertigt ist, wenn sie die Nutzenden eigentlich schützen soll, aus Sicht des Verbraucher- und Umweltschutzes aber eigentlich nachteilig ist.

In der vergleichenden Fallanalyse konnte weiter festgestellt werden, dass Hersteller bisher nur unzureichend über Software-Änderungen informieren, die einen Einfluss auf die Funktionalität und Performance des Gerätes haben können – eine Praxis, die in einigen Ländern bereits zu erfolgreichen Klagen gegen Hersteller wegen Konsument*innen-Täuschung geführt hat. Hersteller sind bisher relativ frei, was die Ausgestaltung der Supportzeiträume für Software angeht. Eine vergleichende Betrachtung des Smartphone-Marktes zeigt, dass die Zeiträume für den Softwaresupport generell sehr unterschiedlich sind. In einem spezifischen Fall konnte gezeigt werden, dass ein Hersteller die Unterstützung für eine Cloud-Plattform innerhalb eines Monats nach Ankündigung eingestellt hat, ohne den Nutzer*innen eine hinreichende Begründung oder Alternativen für die mit dem Betrieb verbundenen Geräte anzubieten.

Obwohl der Einfluss von Software bei Unterhaltungs- und Haushaltsprodukten stark zunimmt, handelt es sich um eine relativ junge Marktentwicklung. Im Rahmen der Untersuchung konnten bisher nur wenige stichhaltige Fälle ermittelt werden, bei denen softwarerelevante Aspekte die Wahrscheinlichkeit für die Obsoleszenz des Produkts erhöhen. Ein Befund, der ebenfalls von den Ergebnissen der repräsentativen Befragung im Rahmen der Studie gestützt wird (siehe Kapitel 6.1.2). Die Fallstudien belegen jedoch auch, dass die Ausgestaltung und Zusicherung von funktionsrelevanten Softwareeigenschaften bisher schlecht reguliert und weitestgehend den Herstellenden überlassen sind, woraus eine erhebliche Informationsasymmetrie gegenüber Nutzer*innen entsteht. Diese werden in der Regel wenig darüber informiert, welche Auswirkungen Softwareänderungen auf die Geräteperformance haben und wie lange sie mit einer hochwertigen Softwareunterstützung rechnen können.

Wahrnehmungen und Erfahrungen von Konsument*innen: Softwareobsoleszenz als kommendes Problem

Im Projekt sollte festgestellt werden, wie häufig Nutzer*innen smarterer Geräte in ihrem Alltag tatsächlich Erfahrungen mit Fällen von softwarebedingter Obsoleszenz erleben und wie sie damit umgehen. Hierzu wurden zwei Datenquellen genutzt und ausgewertet: Zum einen wurden Nutzer*innen-Foren und Produktrezensionen im Internet untersucht, in denen selbsterlebte Fälle von softwarebedingter Obsoleszenz berichtet und diskutiert werden. Zum anderen wurde eine repräsentative Umfrage unter 1.000 Personen zwischen 18 und 75 Jahren per Online-Fragebogen zu aktuell im Haushalt eingesetzten smarten Elektronikprodukten, geplanten Anschaffungen, wahrgenommenen Verschlechterungen sowie Praktiken im Umgang mit Problemen und Erwartungen an Produkte durchgeführt.

Die Auswertung der Forenbeiträge und Produktrezensionen zeigen, dass smarte Geräte, die nur in Verbindung mit Server-Infrastruktur funktionieren oder auf Smartphone-Apps zur Steuerung angewiesen sind, zum Teil spontan unbenutzbar werden, in Einzelfällen bereits kurz nach dem Kauf. Aus Sicht der Nutzer*innen kommunizieren der Hersteller bzw. Anbieter jedoch nicht ausreichend transparent die Gründe für den Ausfall. Zudem wird vermutet, dass häufig technische Gründe für softwarebedingte Obsoleszenz vorgeschoben werden, die von außen nicht überprüfbar und für Nutzer*innen somit unbefriedigend sind. Nutzer*innen berichten, dass smarte Geräte oft besonders gut im Zusammenspiel mit anderen Geräten desselben Herstellers funktionieren. Dadurch können jedoch Lock-In-Effekte entstehen, welche die Kompatibilität mit anderen Systemen verringert und zu einer Herstellerabhängigkeit führt. Zudem befürchten Nutzer*innen die Obsoleszenz einer großen Anzahl von Geräten im Haushalt, wenn sich Standards und Schnittstellen verändern.

Während die Informationen aus den Online-Foren und Produktrezensionen aufschlussreiche Einblicke in spezifische Wahrnehmungen und Erlebnisse geben, kann eine Repräsentativbefragung zeigen, wie häufig softwarebedingte Obsoleszenz mit welchen Konsequenzen erlebt wird. Die im Projekt umgesetzte Befragung zeigt, dass Softwareobsoleszenz (noch) kein weit verbreitetes Problem ist. Dies kann zum einen daran liegen, dass die am stärksten verbreiteten digitalen bzw. vernetzten Geräte, wie Smartphones, Internet-Router, PCs und Smart-TVs oft nur einige wenige Jahre genutzt und häufig ausgetauscht werden, bevor größere Probleme auftreten. Die Umfrage zeigt jedoch auch, dass andere – in der Regel eher lange genutzte und langlebige - Produkte wie Saugroboter, Beleuchtung und Sicherheitsprodukte die am häufigsten geplanten Anschaffungen smarterer Geräte sind. Das bedeutet, dass sich das Problem softwarebedingter Obsoleszenz ähnlich wie auf Basis der Marktentwicklungsanalyse angenommen, in Zukunft eher verstärken wird.

Mit den aktuell genutzten vernetzten Geräten ist die Mehrzahl der Befragten zufrieden, Verschlechterungen werden eher in Einzelfällen und insbesondere bei SmartTVs und Saugrobotern wahrgenommen, wobei Verbindungsprobleme am häufigsten sind. In den meisten Fällen wird das Produkt trotz Verschlechterung weiter genutzt. Auffällig waren in der Befragung vor allem Altersunterschiede. Jüngere Befragte nehmen Verschlechterungen smarterer Geräte über die Benutzungszeit häufiger wahr als ältere Befragte. Während jüngere Nutzer*innen eher online nach Lösungen für Probleme mit Geräten suchen, fragen ältere Nutzer*innen eher Familienmitglieder und Bekannte.

Eine Mehrheit der Befragten wünscht mehr Informationen zu Haltbarkeit und Software-Unterstützung von Geräten, mehr Wissen zur sachgemäßen Pflege und Wartung, mehr Unterstützung bei Fehlern und Einschränkungen und mehr Informationen durch den Handel und die Hersteller. Ca 50 % der Befragten wünscht sich zudem mehr einklagbare Rechte für Nutzer*innen und mehr Informationen über die Umweltbelastungen.

Für die überwiegende Mehrheit der Befragten liegt die Verantwortung für die Software beim Hersteller der Geräte. Ein geringer Teil der Befragten sieht sich selbst als Nutzer*in verantwortlich. Handel sowie Gesetzgeber/ Politik werden hingegen kaum als wichtige Einflussgruppen im Kontext softwarebedingter Obsoleszenz gesehen.

Einschätzungen von Expert*innen: Einigkeit über Risiken, Uneinigkeit bei den Lösungen

Da das Thema softwarebedingte Obsoleszenz verschiedene Bereiche der Gesellschaft betrifft und ein ganzheitlicher Ansatz zur Lösung dieser Herausforderung nötig ist, wurden im Rahmen des Vorhabens unterschiedliche Akteursgruppen über die Einrichtung eines Begleitkreises einbezogen. Ziel der Begleitkreistreffen die gemeinsame Diskussion mit Expert*innen, die ihre spezifischen Erfahrungen und Kenntnissen in die Erarbeitung der Ergebnisse einbringen sollen. Zudem wurden Personen aus dem Begleitkreis im Rahmen vertiefender Interviews befragt, um den Stand des Wissens zu softwarebasierter Obsoleszenz zu erfassen – einerseits zu Ursachen und Entstehung von softwarebedingter Obsoleszenz und andererseits zu Einflussmöglichkeiten von Seiten der Hersteller, Konsument*innen und der Politik. Der Begleitkreis setzte sich zusammen aus Expert*innen verschiedener gesellschaftlicher Bereiche und Sektoren (Forschung, Unternehmen und Start-Ups im Software- und Elektronik-Bereich, Branchenverbände, Vertreter*innen aus Zivilgesellschaft und Open Source-Bewegung).

Das Thema softwarebedingte Obsoleszenz wird von den unterschiedlichen Akteuren im Bereich IKT als zunehmend relevant wahrgenommen und bearbeitet. Dabei gibt es bisher keine allgemein genutzte Definition des Begriffes softwarebedingte Obsoleszenz, so dass unterschiedliche Mechanismen und Vorgänge darunter verstanden werden. Einigkeit besteht zwischen den Expert*innen darüber, dass Obsoleszenzrisiken durch geeignetes Vorgehen während Produktplanung, -entwicklung und Support verringert werden könnten. Aus Sicht der befragten Akteure liegt die Vermeidung von softwarebedingter Obsoleszenz nicht in der Hand Einzelner, sondern muss als Organisationsthema verstanden werden.

Gerade bei Consumer-Produkten stehen die unmittelbaren Kosten beim Erwerb bei Kaufentscheidungen im Vordergrund, während die Lebenszykluskosten weniger betrachtet werden. Sorgfältig entwickelte Produkte mit geringerem Obsoleszenzrisiko und erhöhter Haltbarkeit sind aber typischerweise mit höheren Anschaffungskosten verbunden und erst bei Betrachtung der längeren möglichen Nutzungsdauer preislich vorteilhaft.

Mögliche Siegel für gute Produkte, gesetzliche Mindestlebensdauern für Produktklassen, Verbesserungen der Reparierbarkeit von Produkten mit Softwareanteil, Mietmodelle als Alternative zum Gerätekauf und eine Offenlegungspflicht von Quellcode nach Supportende wurden von den Befragten als Lösungsstrategien angesprochen. Über konkrete

Vorgehensweisen und geeignete politische Anreize, die zu guter Software mit geringem Obsoleszenzrisiko führten, besteht jedoch keine Einigkeit.

Produktpolitische Maßnahmen gegen Softwareobsoleszenz: Marktzugang regeln, zu Transparenz verpflichten und Öko-Innovation stimulieren

Die verschiedenen Untersuchungen im Projekt haben verdeutlicht, dass Software einen wesentlichen Einfluss auf die Nutzungsdauer von vernetzten und mikroelektronischen Produkten hat. Softwarebedingte Obsoleszenz kann zu einem erhöhten Aufkommen an Elektronikschrott, einem erhöhten Ressourcenverbrauch zur Herstellung von Neugeräten und zu wirtschaftlichen Belastungen für die Anwender*innen bzw. Konsument*innen solcher Produkte führen. Einer softwarebedingten Obsoleszenz sollte deshalb politisch entgegengesteuert werden. Ziel von produktpolitischen Maßnahmen ist es in diesem Fall, eine möglichst lange Nutzungsdauer von Produkten zu ermöglichen. Die Maßnahmen sollten sowohl die Hersteller und Händler von Produkten adressieren, aber auch die für die Funktionalität der Produkte erforderlichen Dienstleister (z.B. Cloud-Service-Anbieter, Softwareentwickler*innen) und die Konsument*innen selbst, die letztlich über die Entsorgung und den Neukauf von Produkten entscheiden.

Die softwarebedingte Einflussnahme auf die Verkürzung der Nutzungsdauer von Produkten wird bisher nur in einigen Facetten und insbesondere von unterschiedlichen regulatorischen Vorgaben adressiert. Die Warenkaufrichtlinie (EU 2019/771) stellt bereits Anforderungen an Informationspflichten und den Erhalt der Vertragsmäßigkeit von Produkten sowie dessen Qualität, was auch die Haltbarkeit und Bereitstellung von Software umfasst. Zudem ist in Deutschland ein Gewährleistungszeitraum von zwei Jahren festgelegt. Die Ökodesign-Richtlinie (RL (EU) 2009/125/EG) legt bereits Mindestanforderungen an den Marktzugang für energieverbrauchsrelevante Produkte fest und enthält für einige Produkte bereits konkrete Anforderungen an die Mindestverfügbarkeit von Soft- und Firmware sowie den Zugang für reparaturrelevante Informationen.

Softwarebedingte Obsoleszenz kann in all ihren Aspekten regulatorisch eingedämmt werden, da die Ursachen meist organisatorischer und wirtschaftlicher Art sind und nicht technisch-physikalischer Art. Die nachfolgenden Empfehlungen richten sich nicht nur an die nationale Gesetzgebung. Da viele Akteure global agieren, sollte mindestens EU-weit ein level-playing field geschaffen werden.

Die zwölf empfohlenen Handlungsstrategien sind nach den drei Marktmechanismen Mindestanforderungen für den Marktzugang, Transparenzmaßnahmen zur Förderung des Wettbewerbs und Förderung von Öko-Innovationen gruppiert. Transparenzmaßnahmen sind dabei von besonderer Bedeutung, da sie nicht nur die Beschaffung und Kund*innen befähigen, eine informierte ökologische Entscheidung zu treffen, sondern auch elementarer Baustein sind, auf dem die anderen beiden Maßnahmenpakete aufbauen können. Zudem begegnen Transparenzmaßnahmen in der Umsetzung meist weniger Barrieren als Mindestanforderungen. So kann beispielsweise Maßnahme 8 (Angabe des garantierten Supportzeitraums) sofort eingeführt und im zeitlichen Verlauf durch Maßnahme 3 (Gewährleistung einer Mindestnutzungsdauer) ergänzt werden.

Die Mindestanforderungen für den Marktzugang umfassen:

1. die Möglichkeit zum Betrieb der Geräte ohne externe Abhängigkeiten,
2. die Bereitstellung von sicherheitsrelevanten Software-Updates über einen Mindestzeitraum von 10 Jahren,
3. die Gewährleistung einer Mindestnutzungsdauer von typischerweise 10 Jahren,
4. das Verbot von softwareseitigem Verhindern von Reparatur und Betrieb,

5. Anforderungen an die Kompatibilität und Interoperabilität von Produktsystemen,
6. Anforderungen an externe Dienstleistungen zur Aufrechterhaltung ihrer Dienste für mindestens 10 Jahre sowie
7. Anforderungen an Verkaufs- und Vertriebsplattformen von Software/Apps

Die Transparenzmaßnahmen zur Förderung des Wettbewerbs werden gewährleistet durch:

8. verpflichtende Nennung der Abhängigkeiten von softwarebetriebenen Produkten,
9. verpflichtende Angabe des garantierten Supportzeitraums und daraus resultierende Rechtsansprüche,
10. Kennzeichnung von Produkten zu ihrer Reparierbarkeit sowie
11. verpflichtende Nennung der übertragenen Datenmengen und Art der Übertragungsinhalte.

Öko-Innovationen werden gefördert durch:

12. eine verpflichtende umweltverträgliche öffentliche Beschaffung und
13. die Förderung und Lehre nachhaltiger Softwareentwicklung.

Die im Projekt erarbeiteten Handlungsempfehlungen können sowohl in bereits existierende Gesetze integriert werden als auch z.B. im Rahmen des Circular Economy Action Plans (CEAP) in einer eigenen Verordnung oder Richtlinie münden.

Weiterer Forschungsbedarf: Mehr Wissen über Grundlagen sowie ökologische und rechtliche Implikationen nötig

Unter Verwendung vielfältiger Methoden hat diese Studie beispielhaft Probleme softwarebedingter Obsoleszenz definiert, erklärt und mögliche Lösungswege aufgezeigt. Dennoch kann sie weder in der Auswahl der Fallbeispiele noch in den Wirkmechanismen und Lösungspfaden einen Anspruch auf Vollständigkeit erheben. Im Folgenden soll daher der weitere Forschungsbedarf skizziert werden.

Grundlagenforschung:

- ▶ Vertiefende Recherchen und Analysen zu kontextverwandten Konzepten und Begriffen wie beispielsweise der „software evolution“ und dem „software aging“ könnten in Zukunft zu einer besseren Operationalisierung und Formalisierung beitragen. In diesem Zusammenhang sollte eine weitere Auswertung von bestehenden Standards und Modellen stattfinden. Insbesondere im industriellen Bereich gibt es bereits eine Vielzahl an Scoring-Modellen und Bewertungssystemen im Softwareentwicklungsbereich, von dem weitere Regulierungsvorhaben profitieren können. Es sollten Vorschläge entwickelt und Bedarfsklärungen mit den (inter-) nationalen Gremien vorgenommen werden.
- ▶ In dieser Studie konnte nicht gezeigt werden, wie softwarebedingte Obsoleszenz mit anderen Spielarten von Obsoleszenz derselben Produktgruppen wechselwirkt und ob gegebenenfalls Synergieeffekte politischer Maßnahmen entstehen. In diesem Kontext sollten insbesondere globale Lieferketten von Mikrochips und Bereitstellung von Treibersoftware untersucht werden.

Folgenabschätzung:

- ▶ Ein nachfolgendes Forschungsvorhaben könnte den Versuch unternehmen, die Umwelteffekte und wirtschaftlichen Folgekosten, sowie Externalisierungs- und Verteilungseffekte softwarebedingter Obsoleszenz zu quantifizieren.
- ▶ Weiterhin sollte der Frage nachgegangen werden, ob Unternehmen ökonomische Vorteile erwachsen können, wenn sie Obsoleszenz vermeiden, oder ob ein grundsätzlicher

Zielkonflikt zwischen Gewinnerwirtschaftung und Ökologie im Design lang haltbarer Produkte besteht. Darauf aufbauend stellt sich die Frage, wie Gesellschaft und Regierungen angesichts multipler ökologischer Problemlagen mit diesem Zielkonflikt umgehen wollen.

Politischer und rechtlicher Rahmen:

- ▶ Im Rahmen der vorliegenden Studie war es nicht möglich, eine genaue rechtliche Bewertung und Sammlung von Fällen von Kund*innentäuschung und Wettbewerbsverzerrung im Kontext softwarebedingter Obsoleszenz vorzunehmen.
- ▶ Es konnte nicht abschließend geklärt werden, welche Pflichten aus der Warenkaufrichtlinie (EU 2019/771) erwachsen. Dafür wäre eine Definition und Auslegung von Verkehrssitte und von vernünftigerweise erwartbaren Qualitätsanforderungen nötig.
- ▶ Eine spannende Frage bleibt, inwieweit die politischen Maßnahmen einzeln und in welcher Konstellation des zeitlichen Inkrafttretens welche Wirksamkeit entfalten.

Des Weiteren soll diese Studie einen Beitrag leisten, Entscheidungsträger*innen und Öffentlichkeit für die ökologischen Folgen der Digitalisierung zu sensibilisieren. Auch hierbei besteht noch Forschungs- und Handlungsbedarf. Offene Fragen sind u.a., wie die bestehende Transformationsforschung im Bereich der Digitalisierung auch für das Thema Softwareobsoleszenz sensibilisiert und wie die Thematik in anderen Politikfelder verankert werden kann. Außerdem sollten anschlussfähige Narrative für nicht software-affine Zielgruppen entwickelt werden.

Summary

The proportion of devices in everyday use that are dependent on software is steadily increasing. Not only in classic digital devices such as smartphones and laptops, but also in the field of consumer electronics, the proportion of devices whose functionality is enabled by integrated microcontrollers and associated software is increasing. This trend is increasingly spreading to household appliances, building technology systems and vehicles. Therefore, the quality of the software also determines the service life, functionality, and reliability of devices to an increasing extent. On the one hand, software can be used to define or change the useful life and ease of use of devices. On the other hand, the security of use and the continuity of compatibility between hardware and software as well as between networked devices remain a constant challenge. For long-lived products, software-related obsolescence can become a significant problem for the durability and reliability of the devices.

From an environmental perspective, the problem is obvious and is already being addressed in some cases by current strategies in national and European legislation, such as the Environmental Digital Agenda (BMU 2020) or the European Circular Economy Action Plan (EC 2020). In the field of social-ecological research, the phenomenon of software obsolescence is still insufficiently investigated. However, the public attention for the topic as well as the growing understanding of the problem by various societal stakeholders shows the growing need for systematization and research of the causes, phenomena and problem-solving approaches related to software-related obsolescence.

Against this background, the German Federal Environment Agency launched the project "Analysis of software-induced shortening of the useful life of products " (in short: "SobO"; funding code: UFOPLAN 3719 37 309 0) at the end of 2019. The aim of the project was to investigate systematically the manifestations and causes of software-related shortened useful and service lives and to derive political options for action.

In order to identify effective political strategies, various key sectors (consumption and consumer protection, production and product testing, hardware and software development, political framework conditions) were examined in their interplay, incorporating interdisciplinary and transdisciplinary perspectives and drawing on a broad range of expertise from the social sciences, engineering, economics and political science. Methodologically, in addition to qualitative and quantitative surveys, various empirical case studies, a market and trend analysis, and in-depth cause-and-effect analyses of networked device systems were applied. In the following, the central results are summarized along the chapter structure.

Definition of software-related obsolescence: Consider the dynamics of the product system!

A formal definition of software obsolescence has been lacking so far, and the phenomenon is currently the subject of little scientific research. Although various relations and causes have already been identified in the existing literature, a general definition of the object of research and the problem area is still lacking. The situation is further complicated by the fact that software is not subject to any directly observable physical wear and tear, as is the case with tangible goods, for example, whose service life is primarily limited by permanent thermal, oxidative, and physical stresses. Rather, software obsolescence arises within the dynamic arrangements of a product system, within which software performs or can no longer perform specific functions. Software obsolescence is a collective term for phenomena in which the quality of software degrades to a degree where the product system associated with it no longer adequately fulfills its intended function for a significant number of users. Software obsolescence is triggered by the software itself (directly) or changing requirements for the functional and

non-functional properties of the software (indirectly). If product systems connected with software become obsolete, the software can play a more or less strong role as a possible factor. In order to better assess the functionality or possible risk factors associated with the software of a product system, the so-called LEUF scheme (see chapter 1.3.2) was developed in the project. Along specific quality and functional conditions (Legal, Executable, Usable, Functional), risk factors for software obsolescence in application and use are to be better formalized and mapped in a simple chain of events.

State of the art: Determination of software quality as a research gap

Software obsolescence is not a new phenomenon and can be found along the entire computer and software history, which took its starting point in the 1940s with the first development of non-civil mainframes (computers). The project's literature review has revealed that the topic of software-related obsolescence is already addressed within the professional community in various discourses, but the problem itself has only been studied and formalized to a small extent. In general, the various manifestations or partial aspects of software-related obsolescence are treated under different terms such as "software rot", "software erosion", "software aging". Regardless of context and mode of operation, all concepts share the assumption that the long-term functionality of software can be made dependent on certain quality characteristics, which can, however, change and degrade over time as a result of changing internal and external requirements. The ICT industry has since been characterized by short market cycles, a high pace of innovation and increasing diversification of software and hardware technologies, making it difficult to generalize and standardize relevant software qualities. The literature review identified central quality standards and norms such as ISO/IEC 25010 for determining software product quality, but their actual application and implementation in practice remain questionable and require more in-depth empirical investigations. However, the inclusion of minimum requirements and criteria for the quality of software in individual Ecodesign Directives (Directive 2009/125/EC) since 2021, as well as the addition of information requirements for digital elements to the Purchasing of Goods Directive (EU 2019/771), demonstrate the growing need for greater scientific formalization and practical operationalization of the use- and lifetime-relevant properties of software.

Interdependencies of software and hardware: A question of compatibility, functionality and security

In order to better understand the cause-effect relationships in the context of software-related obsolescence, a comparative analysis of different product systems was carried out based on the OSI layer model. The OSI layer model (Open Systems Interconnection Model) is a widely used model for describing software architectures. For the analysis, in addition to the functional dependencies within the OSI layer model, business practices and the interdependencies of the various actors (software and hardware developers, sales platforms, users) were also investigated, from which obsolescence risks arise in practice. Three central impact categories were identified for the long-term usability of software-based and software-dependent products, which were examined in more detail on the basis of empirical cases. This results in three directions to curb software-related obsolescence:

- ▶ Long-term interoperability and compatibility between different devices, device generations, and standards.
- ▶ Provision and long-term guarantee of defined core functionalities for products within a product system
- ▶ Maintaining security through concepts such as security by design and high-quality software.

Software-based and software-dependent products in networked product systems are generally subject to permanent changes such as bug fixes or security and function updates. Analysis shows that the various actors along the software supply chain have particular powers of influence which they can use to the detriment of other actors (key position of gatekeepers). This includes, for example, the withholding of important security and function updates or compatibility with existing interfaces. The role of the gatekeeper does not always lie with the manufacturer or distributor of the products, but also with platform operators (e.g. contract conditions in sales portals) and other third parties in the software supply chain (e.g. provision period for firmware for chipsets or operating systems). Their indirect product responsibility should be investigated more closely and addressed in regulatory terms.

Market trends and development: SmartHomes on the rise

In order to better assess future developments and environmental relevance, the project examined the market development of household appliances up to the year 2030. The focus was on smart home product systems, with smartphones and smart TVs being two particularly dynamic product segments that are relevant due to their high penetration.

The study of household development in Germany shows a market potential of 837,000 new one- and two-person households for the smart home market by 2030. The market participants can basically be divided into module developers, sensor manufacturers and gateway developers. Companies such as Phillips, OSRAM, SIEMENS and Schneider Electric are most strongly positioned in module development. In addition, Google, Amazon, Apple and Microsoft play a dominant role in current and future market developments through their gateway applications. For the product groups considered, it can be stated that in the smart TV segment, the operating systems AndroidTV, Tizen and WebOS are considered to be the main technological drivers with a share of 81.9% of the devices on sale in the German-speaking region. In the smartphone segment, the four manufacturers Samsung, Apple, Huawei and Xiaomi occupy 84% of the European market. Compared to the German smartphone market, which is already showing the first signs of saturation, the penetration rate on the European market was 77.6% in 2018 (Statista 2021).

For the final assessment, a rating system is presented based on seven indicators, which enables a rough estimate of software-related obsolescence risks of systems and platforms. The rating system is based on the impact categories already identified in advance: compatibility, functionality and security of software.

Analysis of empirical cases of software-related obsolescence: The big transparency problem

Among smart electronics currently used in households (household appliances, ICT, consumer electronics), various cases of software-related obsolescence were identified for this study and subjected to an exploratory case analysis. A total of five empirical cases were examined:

- ▶ Software-controlled counters in printers
- ▶ Trade-up programs in Bluetooth speakers using software-controlled shutdown devices ("kill switch")
- ▶ Software-based performance degradation in smartphones
- ▶ Support period and end of support for smartphone software
- ▶ End-of-life of cloud-based platforms.

In the study, the case analyses serve to validate and expand the underlying mechanisms and develop practice-oriented recommendations for action.

In the individual case analysis, it is noticeable that manufacturers justify software-based function restrictions and configurations predominantly with reference to quality assurance measures and product safety. Examples include counters and consumption meters for printer cartridges to ensure optimal printing results, software-based serialization of individual components to prevent the installation of spare parts from third-party suppliers, and software-based performance throttling to extend battery life. In one case, a software-controlled "kill switch" was used to render used products completely inoperable with the consent of the respective user in order to receive a voucher for the purchase of a new product ("trade-in"). It can be stated that the appropriate use of protective devices by software has so far been a legal grey area and is predominantly at the discretion of the manufacturer. In some cases, however, this can cause the premature loss of functionality of hardware. It is therefore necessary to clarify in greater depth how much protection against overuse and misuse a function-limiting intervention by software is actually justified from the point of view of consumer and environmental protection.

The comparative case analysis also revealed that manufacturers have so far provided insufficient information about software changes that could affect the functionality and performance of the device - a practice that has already led to successful lawsuits against manufacturers for deceiving consumers in some countries. Manufacturers have so far been relatively free when it comes to the design of support periods for software. A comparative analysis of the smartphone market shows that the time periods for software support are generally very different. In one specific case, it was shown that a manufacturer discontinued support for a cloud platform within a month of announcement without offering users sufficient justification or alternatives for the devices associated with its operation.

Despite the growing influence of software in electronic products, the research identified few valid cases where software-related issues increased the likelihood of product obsolescence. A finding which is also supported by the results of the representative consumer survey (see section 6.1.2). However, the case studies also show that the design and assurance of functionally relevant software properties have been poorly regulated to date and are largely left to the manufacturers, resulting in considerable information asymmetry vis-à-vis users. Users are usually given little information about the effects of software changes on device performance and how long they can expect high-quality software support.

Perceptions and experiences of consumers: Software obsolescence as a coming problem

The project aimed to determine how often users of smart devices experience cases of software-related obsolescence in their everyday lives and how they deal with them. Two data sources were used and evaluated for this purpose: Firstly, user forums and product reviews on the internet were examined, in which self-experienced cases of software-related obsolescence are reported and discussed. Secondly, a representative online survey of 1,000 people between the ages of 18 and 75 was conducted on smart electronic products currently used in the household, planned purchases, perceived deterioration, and practices in dealing with problems and expectations of products.

The analysis of forum posts and product reviews show that smart devices that only work in conjunction with server infrastructure or rely on smartphone apps for control sometimes spontaneously become unusable, in individual cases shortly after purchase. From the users' point of view, however, the manufacturer or provider does not communicate the reasons for the failure with sufficient transparency. In addition, it is suspected that technical reasons for

software-related obsolescence are often put forward, which cannot be verified externally and are therefore unsatisfactory for users. Users report that smart devices often work particularly well in combination with other devices from the same manufacturer. However, this can lead to lock-in effects and reduced compatibility with other systems. In addition, users fear the obsolescence of a large number of devices in the household if interfaces change.

While information from online forums and product reviews provide insight into specific perceptions and experiences, a representative survey can show how often software-related obsolescence is experienced and with what consequences. The survey implemented in the project shows that software obsolescence is not (yet) a widespread problem. On the one hand, this may be due to the fact that the most widespread digital or networked devices, such as smartphones, Internet routers, PCs and smart TVs, are often only used for a few years and are frequently replaced before major problems occur. However, the survey also shows that other products - which tend to be more long-used and long-lasting - such as robot vacuums, lighting and security products are the most commonly planned smart device purchases. This means that similar to what was assumed on the basis of the market development analysis, the problem of software-related obsolescence is more likely to intensify in the future.

The majority of respondents are satisfied with the networked devices currently in use; deterioration is perceived more in individual cases and particularly in smart TVs and vacuum robots, with connectivity problems being the most common. In most cases, the product continues to be used despite deterioration. Age differences were particularly noticeable in the survey. Younger respondents perceive deterioration of smart devices over the period of use more frequently than older respondents. While younger users are more likely to look online for solutions to problems with devices, older users are more likely to ask family members and acquaintances.

A majority of respondents want more information on durability and software support for devices, more knowledge on proper care and maintenance, more support for faults and limitations, and more information from retailers and manufacturers. Approximately 50% of respondents also want more enforceable rights for users and more information about environmental impacts.

For the vast majority of respondents, the responsibility for the software lies with the manufacturer of the devices. A small number of respondents see themselves as responsible users. Retailers and policy makers, on the other hand, are hardly seen as important influence groups in the context of software-related obsolescence.

Assessments by experts: Agreement on risks, disagreement on solutions

Since the topic of software-related obsolescence affects various areas of society and a holistic approach to solving this challenge is necessary, various stakeholder groups were involved in the project through the establishment of an external expert board. The aim of the board meetings was the joint discussion with experts, who were to contribute their specific experience and knowledge to the elaboration of the results. In addition, people from the board were questioned in in-depth interviews in order to record the state of knowledge about software-based obsolescence - on the one hand, about the causes and emergence of software-based obsolescence and, on the other hand, about the possibilities for influence on the part of manufacturers, consumers and politicians. The expert board consisted of experts from various social areas and sectors (research, companies and start-ups in the software and electronics sector, industry associations, representatives from civil society and the open source movement).

The topic of software-related obsolescence is increasingly being perceived and addressed by the various players in the ICT sector. To date, there is no generally used definition of the term

software-related obsolescence, so that different mechanisms and processes are understood to be involved. There is agreement among experts that obsolescence risks could be reduced by appropriate procedures during product planning, development, and support. From the point of view of the stakeholders interviewed, the avoidance of software-related obsolescence is not in the hands of individuals but must be understood as an organizational issue.

Particularly in the case of consumer products, the immediate costs of acquisition are at the forefront of purchasing decisions, while the lifecycle costs are considered to a lesser extent. Carefully developed products with a lower risk of obsolescence and increased durability, however, are typically associated with higher acquisition costs and are only advantageous in terms of price when the longer possible useful life is taken into account.

Possible labels for good products, statutory minimum service lives for product classes, improvements in the reparability of products with a software component, rental models as an alternative to purchasing equipment, and mandatory disclosure of source code after the end of support were mentioned by the interviewees as solution strategies. However, there is no agreement on concrete approaches and suitable political incentives that would lead to good software with low obsolescence risk.

Product policy measures against software obsolescence: Regulate market access, oblige transparency and stimulate eco-innovation

The various studies in the project have made it clear that software has a significant influence on the useful life of networked and microelectronic products. Software-induced obsolescence can lead to an increased amount of electronic waste, an increased consumption of resources for the production of new devices and to economic burdens for the users and consumers of such products. Therefore, software-related obsolescence should be counteracted politically. In this case, the aim of product policy measures is to enable products to have the longest possible useful life. The measures should address both the manufacturers and retailers of products, but also the service providers required for the functionality of the products (e.g. cloud service providers, software developers) and the consumers themselves, who ultimately decide whether to dispose of products or buy new ones.

The software-related influence on the shortening of the useful life of products has so far only been addressed in some facets and in particular by different regulatory requirements. The Sale of Goods Directive (EU 2019/771) already sets requirements for information obligations and the preservation of contractual compliance of products as well as its quality, which also includes the durability and provision of software. In addition, a warranty period of two years is specified in Germany. The Ecodesign Directive (RL (EU) 2019/125/EC) already specifies minimum market access requirements for energy-related products and, for some products, already contains concrete requirements for the minimum availability of software and firmware as well as access for repair-related information.

Software-related obsolescence can be curbed in all its aspects by regulation since the causes are mostly organizational and economic rather than technical and physical. The following recommendations are not only directed at national legislation. Since many actors act globally, a level-playing field should be created at least across the EU.

The ten recommended action strategies are grouped according to the three market mechanisms: minimum requirements for market access, transparency measures to promote competition, and promotion of eco-innovation. Transparency measures are of particular importance, as they not only enable procurement and customers to make informed environmental choices but are also elementary building blocks on which the other two sets of measures can build. In addition, transparency measures usually face fewer barriers to implementation than minimum

requirements. For example, Measure 7 can be introduced immediately and supplemented over time by Measure 3.

Minimum requirements for market access include:

1. the ability to operate the devices without external dependencies,
2. the provision of security-related software updates over a minimum period of 10 years,
3. the guarantee of a minimum useful life of typically 10 years,
4. the prohibition of software-based prevention of repair and operation,
5. requirements for compatibility and interoperability of product systems,
6. requirements for external services to maintain its services for at least 10 years, and
7. requirements for sales and distribution platforms of software/apps.

Transparency measures to promote competition shall be ensured by:

8. mandatory citing dependencies on software-driven products,
9. a mandatory indication of the guaranteed support period and resulting legal rights,
10. labelling products on their reparability, and
11. obligatory naming of the amount of data transmitted and the type of transmission content.

Eco-innovations are stimulated by:

12. a mandatory green public procurement
13. the promotion and education of sustainable software development

The recommendations for action developed in the project can be integrated into existing legislation or, for example, result in a separate regulation or directive as part of the Circular Economy Action Plan (CEAP).

Further research needed on conceptual basis of software obsolescence as well as ecological and legal implications of regulations

Based on a multi-method approach, this study has defined and explained exemplary problems of software-related obsolescence and identified possible solutions. Nevertheless, it cannot claim to be exhaustive either in its selection of case studies or in its analysis of causes and solution paths. Therefore, the need for further research will be outlined in the following.

Basic research:

- ▶ In-depth research and analysis on contextually related concepts and terms such as "software evolution" and "software aging" could contribute to better operationalization and formalization in the future. In this context, further evaluation of existing standards and models should take place. Particularly in the industrial sector, there is already a large number of scoring models and evaluation systems in the software development area from which further regulatory projects can benefit. Proposals should be developed and needs clarifications should be made with the (inter-) national bodies.
- ▶ This study was not able to show how software-related obsolescence interacts with other varieties of obsolescence of the same product groups and whether, if so, synergy effects of policy measures arise. In this context, global supply chains of microchips and provision of driver software in particular should be investigated.

Impact Assessment:

- ▶ A subsequent research project could attempt to quantify the environmental effects and economic follow-up costs, as well as externalization and distribution effects of software-related obsolescence.
- ▶ Furthermore, the question should be investigated whether companies can gain economic advantages by avoiding obsolescence or whether there is a fundamental conflict of objectives between profit-making and ecological impact in the design of long-lasting products. Based on this, the question arises how society and governments want to deal with this trade-off in the face of multiple ecological problems.

Policy and legal framework:

- ▶ In the context of this study, it was not possible to make a precise legal assessment and collection of cases of customer deception and distortion of competition in the context of software-related obsolescence.
- ▶ It was not possible to conclusively clarify which obligations arise from the Sale of Goods Directive (EU 2019/771). This would require a definition and interpretation of custom and reasonably expected quality requirements.
- ▶ An intriguing question remains to what extent the individual or combined policy measures develop which effectiveness.

Furthermore, this study is intended to make a contribution to sensitizing decision-makers and the public to the ecological consequences of digitalization. Here, too, there is still a need for research and action. Open questions include how existing transformation research in the area of digitalization can also be used to raise awareness of the topic of software obsolescence and how the topic can be anchored in other policy areas. In addition, connectable narratives should be developed for target groups that do not have an affinity for software.

1 Einleitung

1.1 Hintergrund und Relevanz

Der Anteil der alltäglich genutzten Geräte, die von Software abhängig sind, nimmt stetig zu. Sowohl im Bereich IKT, der Unterhaltungselektronik, bei Haushaltsgeräten, der Gebäudetechnik bis hin zu Fahrzeugen – sie alle werden zunehmend von Software gesteuert. Immer häufiger entscheidet daher die Software über Nutzungsdauer, Funktionalität und Zuverlässigkeit von Geräten. Über die Software können zum einen die Nutzungsdauer und der Nutzungskomfort von Geräten definiert oder verändert werden. Zum anderen bleiben die Sicherheit der Nutzung und der Fortbestand der Kompatibilität zwischen Hardware und Software sowie zwischen Geräten untereinander eine konstante Herausforderung. Insbesondere bei langlebigen Produkten kann softwarebedingte Obsoleszenz zu einem bedeutenden Problem für die Haltbarkeit und Zuverlässigkeit der Geräte werden.

Die öffentliche und politische Aufmerksamkeit für das Thema Softwareobsoleszenz ist in den vergangenen Jahren deutlich gestiegen. Aktuelle Urteile und Klagen gegen namhafte Hersteller zeigen zudem, dass es in Bezug auf die Sorgfalts-, Informations- und Aussagepflichten politischen und rechtlichen Handlungsbedarf gibt. Aus der Umweltperspektive ist das Problem offensichtlich und wird bereits durch aktuelle Strategien in der nationalen und europäischen Gesetzgebung vereinzelt adressiert. Hierzu zählen vor allen die Umweltpolitische Digitalagenda (BMU 2020) oder der europäische Aktionsplan zur Kreislaufwirtschaft, der unmittelbar von einem Recht auf ein Update von obsoleter Software spricht (EC 2020). Im Rahmen der Ökodesign-Richtlinien (Richtlinie 2009/125/EG) werden seit Anfang 2021 erstmals verpflichtende Anforderungen an die Verfügbarkeitsdauer von Soft- und Firmware für verschiedene Verbraucherprodukte, wie u.a. Displays, Spülmaschinen oder Waschmaschinen eingeführt. Zudem zeigen politische Initiative, wie z.B. die Einführung des Blauen Engel für Software davon, dass Langlebigkeit und Zuverlässigkeit auch bei digitalen und virtuellen Produkten eine hohe Umweltrelevanz haben.

Im Bereich der sozio-ökologischen Forschung ist das Phänomen der Softwareobsoleszenz bisher wenig erforscht und verstanden. Das wachsende Problemverständnis durch gesellschaftliche Akteure und Interessengruppen zeigt jedoch den wachsenden Bedarf, hier stärker Forschung zu betreiben (siehe 4.4). Erste Umfragen zeigen zudem, dass bereits 30% der Konsument*innen ein elektronisches Gerät aufgrund von Softwareproblemen ersetzt haben (Kantar Emnid/vzbv 2017, Jaeger-Erben/ Hipp 2017).

Vor diesem Hintergrund hat das Umweltbundesamt Ende 2019 das Projekt „Analyse der softwarebasierten Einflussnahme auf eine verkürzte Nutzungsdauer von Produkten“ (Förderkennzeichen: UFOPLAN 3719 37 309 0) ins Leben gerufen.

1.2 Fragestellung und Vorgehen

Das Projekt „Analyse der softwarebasierten Einflussnahme auf eine verkürzte Nutzungsdauer von Produkten“, kurz „SobO“ hatte zum Ziel, systematisch die Erscheinungsformen und Ursachen softwarebedingt verkürzter Nutzungs- und Lebensdauern zu untersuchen und auf dieser Basis politische Handlungsoptionen abzuleiten.

Unter Einbezug inter- und transdisziplinärer Perspektiven und mit einer breiten sozial-, ingenieur-, wirtschafts- und politikwissenschaftlicher Expertise werden verschiedene relevante Sektoren (Konsum und Verbraucherschutz, Produktion und Produkttest, Hardware- und

Software-Entwicklung, politische Rahmenbedingungen) in ihrem Zusammenspiel betrachtet, um fundierte Vorschläge für die Ursachenbekämpfung aufzuzeigen.

Das Vorhaben verfolgte folgende Schwerpunkte:

- ▶ Beschreibung und Analyse des Status Quo: Untersuchung der Ist-Situation von softwarebedingt verkürzter Nutzungs- und Lebensdauer von Elektronikgeräten und deren Ursachen
- ▶ Entwicklung von Szenarien und Ausblick: Entwicklung fundierter Annahmen zur Marktentwicklung von Geräten mit integrierter Software und zu den möglichen Folgen auf die Nutzungs- und Lebensdauer (Haltbarkeit, Zuverlässigkeit, Funktionalität)
- ▶ Durchführung vertiefter Recherche an drei ausgewählten Produktgruppen: vertiefte Fallanalysen für relevante Produktgruppen zu den Ursachenkonstellationen für softwarebedingt verkürzter Nutzungs- und Lebensdauer von Elektronikgeräten sowie effiziente Gegenmaßnahmen
- ▶ Erarbeitung von Handlungs- und Umsetzungsempfehlungen mit dem Fokus auf rechtlichen Möglichkeiten
- ▶ Initiierung von Austausch und fachlichen Dialogen mit relevanten Akteuren für die zukünftige Marktentwicklung

Der vorliegende Bericht beschreibt die Ergebnisse des Forschungsprojekts gegliedert nach inhaltlichen Schwerpunkten. Kapitel 1 führt in das Thema und die Projektschwerpunkte ein. Kapitel 2 bietet einen Überblick über den Stand der Forschung und Entwicklung. Kapitel 3 vertieft die Analyse verschiedener Ursachen und Wirkungszusammenhänge im soziotechnischen System von Software und Hardware anhand von drei Fallbeispielen: (1) Corona Warn App (2) smarte Beleuchtungssysteme und (3) Smarte Heizkörperthermostate. In Kapitel 4 wird die Marktentwicklung im Bereich Smart Home untersucht, um mögliche zukünftige Herausforderungen im Bereich der Softwareobsoleszenz abschätzen zu können. Dabei werden mit Smartphones und Smart TVs zwei besonders dynamische und aufgrund ihrer hohen Verbreitung relevante Produktsegmente vertiefend betrachtet. Kapitel 5 widmet sich der detaillierten Analyse aktueller Fälle von Obsoleszenz, die sowohl medial einige Aufmerksamkeit erregt haben als auch unterschiedliche Fallentwicklungen im Hinblick auf die Interventionen und Reaktionen der Anbieter und Hersteller aufweisen. In Kapitel 6 wird auf die Nutzer*innen-Perspektive eingegangen. Dabei werden zum einen Ergebnisse einer Analyse von Nutzer*innen-Foren zusammengefasst, in denen selbsterlebte Fälle von softwarebedingter Obsoleszenz berichtet und diskutiert werden. Zum anderen zeigen die Ergebnisse einer repräsentativen Umfrage, wie häufig Nutzer*innen bei welchen Geräten Softwareobsoleszenz erleben und wie sie damit umgehen. In Kapitel 7 kommen verschiedene Expert*innen und Praktiker*innen aus verschiedenen Sektoren (Wirtschaft, Wissenschaft, Politik, Zivilgesellschaft) zu Wort, die in unterschiedlichen Positionen und aus unterschiedlichen Perspektiven mit dem Thema Softwareobsoleszenz zu tun haben. In Kapitel 8 werden aus der Zusammenschau aller Ergebnisse produktpolitische Maßnahmen unter Berücksichtigung bestehender Initiativen und Regulationen entwickelt. Kapitel 9 schließt mit der Formulierung weiterer Forschungsbedarfe.

1.3 Gegenstandsbeschreibung: Was ist Softwareobsoleszenz?

Bisher gibt es keine formale Definition von Softwareobsoleszenz und das Phänomen ist derzeit noch wenig wissenschaftlich untersucht. In der Literatur lassen sich jedoch verschiedene zentrale Wirkprinzipien identifizieren:

1. Software wird selbst obsolet und/oder bedingt die Obsoleszenz von anderer Hardware und/oder Software (vgl. Sandborn 2007).
2. Die Qualität und Nutzbarkeit von Software ist von direkten (functional) und indirekten (non-functional requirements/constraints) Eigenschaften und Anforderungen abhängig (vgl. ISO/IEC 25000).
3. Software ist in offenen Systemen sich ändernden Anforderungen an ihre direkten und indirekten Eigenschaften und Anforderungen ausgesetzt (Lehmann 1996/ Taentzer et al. 2019).
4. Die Änderungen oder Nicht-Änderungen aufgrund neuer Anforderung bedingt Softwareobsoleszenz und kann als Alterungsprozess beschrieben werden (Cotroneo et al. 2014; Grottke et al. 2008).

Softwareobsoleszenz tritt diesen Beschreibungen zufolge überwiegend dann ein, wenn bestimmte Funktionalitäten eines Geräts nicht mehr gegeben oder nur noch eingeschränkt vorhanden sind: Sei es die Sicherheit bei vernetzten Geräten, die Zuverlässigkeit bei der Ausführung von Steuerungsaufgaben, die Benutzerfreundlichkeit und Systemperformance oder die Kompatibilität der Software mit anderen Übertragungsstandards, Schnittstellen und Hardwareanforderungen.

1.3.1 Definition und Begriffsabgrenzung

Softwareobsoleszenz kann im Allgemeinen als Verlust von funktionalen (direkten) und nicht-funktionalen (indirekten) Eigenschaften von Software und die mit ihr verbundenen Produkte(-systeme)¹ beschrieben werden.² Eine erste Recherche konnte zeigen, dass innerhalb der Forschung und Fachdiskussionen zum Großteil nur implizit auf Softwareobsoleszenz referiert wird, ohne dabei jedoch explizit den Begriff „Softwareobsoleszenz“ zu benennen.

Softwarebedingte Obsoleszenz und die mit ihr verbundenen Ursachen und Wirkungen werden zum Großteil im Kontext mit verwandten Konzepten und Begriffen wie dem „Software Aging“ beschrieben (siehe Tabelle 1). Innerhalb der Forschung wurde bereits indirekt auf diese Verbindung hingewiesen: „Note that in the case of software engineering, the ‚aging‘, as well as ‚rejuvenation‘ has a different meaning: it indicates the software becoming obsolete, e.g., because of changes requirements, maintenance actions, and so on“ (Cotroneo et al. 2014). Zudem wird auf die Vermeidung vorzeitiger Hardware-Software Obsoleszenz und Ermöglichung langer Nutzungsdauern teilweise auch im Kontext von Designprinzipien wie dem „Green Coding“ und „Nachhaltiger/ressourceneffizienter Software“ eingegangen (Abdullah et al. 2015/ Gröger & Köhn 2015; Hilty et al. 2017; Gröger et al. 2018) auf die im Folgenden jedoch nicht im Detail eingegangen wird. Dies macht deutlich, dass Softwareobsoleszenz bisher kein formal gesicherter Begriff ist und die mit ihm verbundenen Effekte innerhalb der Fachwelt aber bereits in unterschiedlichen Kontexten und Begriffen (Proxys) beschrieben werden.

¹ Siehe Glossar zur Verwendung der Begriffe Produkt und System

² Nicht-funktionale Eigenschaften werden auch als „Qualitätsmerkmale“ oder nicht-funktionale Beschränkungen (non-functional-constraints NFC) beschrieben.

Tabelle 1: Auswahl verwandter Konzepte und Begriffe

Keyword	Relevanz	Beschreibung
software aging	++++	Softwarealterung beschreibt das Phänomen der Zunahme der Fehlerrate oder den Performanceverlust eines laufenden Softwaresystems (Cotroneo et al. 2014, Grottko et al. 2008). Die Alterung hat direkte und indirekte Folgen auf die damit verbundenen Produkte (Parnas 1994).
software aging and rejuvenation (SAR)	+++	Konzept zu Alterung und Verjüngung von Software
software erosion / rot	+++	Umgangssprachliches Konzept zur Beschreibung von Softwarezerfall, -verrottung auch unter ‚bit rot‘ und ‚code rot‘ bekannt. Vergleichbar zum Software Aging.
software blowing / explosion, fat software	+++	Aufblähen von Software in Folge von Updates und der Integration weiterer Features (Wirth 1994/ Parnas 1996)
bug inflation	++	Zunahme der Fehlerrate in Folge von Updates und Änderungen.
software apocalypse / collapse / dependency	++	Problematisierung der Softwareabhängigkeit von Systemen und deren Folgen im Fall von Obsoleszenz (Jang et al. 2017)
software evolution	++	Software Evolution bezeichnet den Lebenszyklus einer Software, die nach Auslieferung und Benutzung aufgrund sich ändernder und neuer Anforderungen verschiedene Anpassungen erfährt (oder nicht) (Lehmann 1996/ Taentzer et al. 2019).

Anders als in der Produktion von Hardware sind die Anforderung an die Softwareausstattung bei vernetzten und offenen Systemen nicht statisch, weil sie sich im Zeitverlauf und während der Nutzung durch Veränderungen der Systemlandschaft dynamisch ändern können. Die Systemlandschaft ändert sich beispielsweise durch neue Applikationen, Updates von Betriebssystemen, neuen Peripheriegeräten, aber auch durch sich ändernde Ansprüche und -gewohnheiten von Nutzenden. Anders als bei physischen Zuverlässigkeits- und Belastungstests handelt es sich bei softwarebedingter Obsoleszenz um einen nicht-trivialen Sachverhalt, der nur unter Einbezug verschiedener direkter und indirekter Faktoren methodisch greifbar gemacht werden kann.

Wir schlagen daher folgende Definition von Softwareobsoleszenz vor:

Arbeitsdefinition für Softwareobsoleszenz

Softwareobsoleszenz ist dann gegeben, wenn die Qualität einer Software bis zu einem Grad abnimmt, bei der das mit ihr verbundene Produktsystem nicht mehr nutzbar ist oder die ihr zuge dachte Funktion nicht mehr hinreichend erfüllt wird. Softwareobsoleszenz wird ausgelöst durch die Software selbst (direkt) oder sich ändernden Anforderungen an die funktionalen und nicht-funktionalen Eigenschaften der Software (indirekt).

Wenn mit Software verbundene Produkte obsolet werden, kann die Software als möglicher Faktor eine mehr oder weniger starke Rolle spielen. In diesem Fall schlagen wir die Differenzierung von zwei verschiedenen Formen von Softwareobsoleszenz vor, die sich aus der Stärke des beobachtbaren Kausalverhältnisses ableiten:

- *Direkte Softwareobsoleszenz*: Die Obsoleszenz wird initiiert durch die Software selbst („Software-induzierte Obsoleszenz“).

- **Indirekte Softwareobsoleszenz:** Die Eigenschaften der Software sind Teil eines Wirkungsgefüges von anderen Faktoren, die eine Obsoleszenz des mit ihr verbundenen Produktsystems bewirken („Software-bedingte Obsoleszenz“).

Beispiele sollen helfen die Unterscheidung greifbar zu machen.

Tabelle 2: Beispiele zur Differenzierung von direkter und indirekter Softwareobsoleszenz

Kausalverhältnis	Beispiele
Direkt	<ul style="list-style-type: none"> • Softwaregesteuerte Abschalt- und Zählvorrichtungen („kill switches“), die eine eingeschränkte Lebensdauer der Produkte zur Folge haben • Fehldesigns, die in Folge höhere Leistungsanforderung zu Hardwaredegradierung führen können („Software bloat“) • Ausbleibende Sicherheitsupdates und Patches zur Fehlerbeseitigung • Funktionale Fehler beim Ausführen der Software
Indirekt	<ul style="list-style-type: none"> • Fehlanpassung der Software durch sich ändernde Systemumgebungen: <ul style="list-style-type: none"> ○ Fehlende Auf- und Abwärtskompatibilität zu Programmen (Apps) und Datenformaten (proprietäre Speicherformate), ○ Inkompatibilität zu neuer oder alter Hardware (Bsp. 5G und anderen Funktechnologien) • Ablauf oder Entzug von Nutzungslizenzen (Abkündigung von Betriebssystemen, Handelskonflikte) • Einstellung des Supports • Verlust oder eingeschränkte Nutzerfreundlichkeit (Usability)

Prinzipiell gilt bei jedem Ursache-Wirkungsgefüge, dass eine Ursache (unabhängige Variable) über ein Kausalverhältnis (Wirkungsrichtung) einen direkten oder indirekten Einfluss auf die Wirkung (abhängige Variable) nehmen kann. In der Fachliteratur zu Obsoleszenz wird dieses Verhältnis unterschiedlich definiert (Pope & Longmuss 2019: 24). Entscheidend ist, dass es sich hierbei um ein Kontinuum handelt, das dabei helfen soll, Bewertungsprobleme aufzudecken.

Direkte Softwareobsoleszenz bietet den analytischen Vorteil, dass sie eindeutig in ihrem Ursache-Wirkungsgefüge beobachtbar und somit auch umweltpolitisch gut adressierbar ist.

Indirekte Softwareobsoleszenz wirkt hingegen weniger offensichtlich und führt nicht immer zum kompletten Funktionsverlust des Produktsystems, was insbesondere für umweltpolitische Maßnahmen eine Herausforderung darstellt. Wenn beispielsweise Nutzungslizenzen für Softwareprodukte auslaufen oder entzogen werden, hindert das in bestimmten Fällen Nutzende nicht daran, diese Produkte weiterhin zu nutzen. Gleiches gilt, wenn bestimmte Apps auf dem Smartphone nicht mehr aktualisiert oder nur noch eingeschränkt nutzbar sind. Indirekte Softwareobsoleszenz ist in diesen Fällen stark abhängig von den Anforderungen der Systemumgebung. Die Systemanforderungen können sich im Zeitverlauf ändern, wozu insbesondere die Erwartungen und -gewohnheiten der Nutzenden zählen. Wenn Betriebssysteme von Smartphones nach einer bestimmten Zeit keine Updates mehr erhalten, dann führt dies nicht automatisch dazu, dass Nutzende ihre alten Geräte gegen Neue austauschen. Es erhöht lediglich die Wahrscheinlichkeit für Obsoleszenz, stellt aber keinen hinreichenden Grund dar.

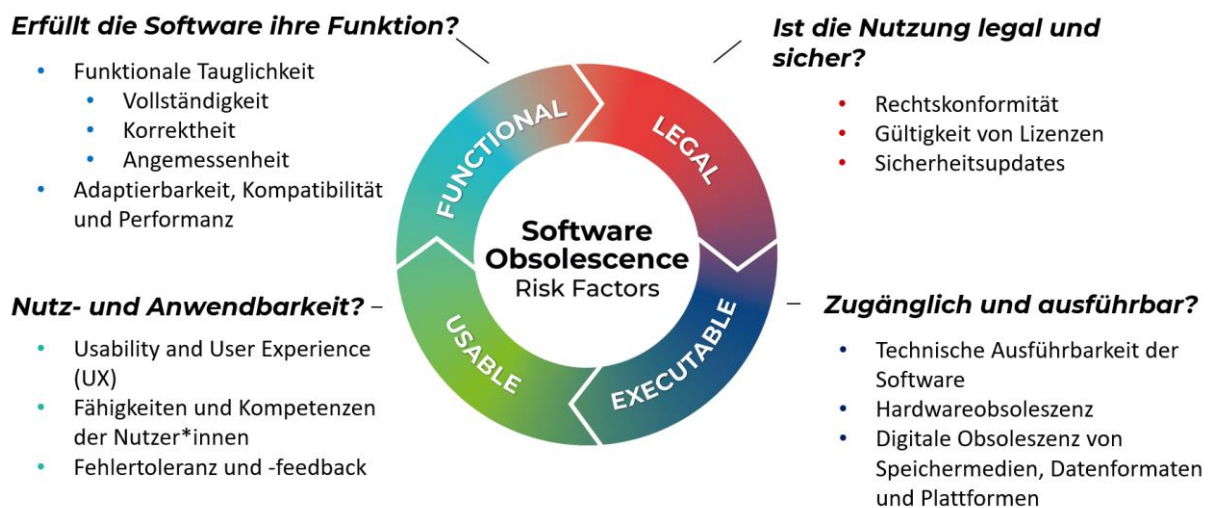
1.3.2 Ursachen von Softwareobsoleszenz (LEUF-Schema)

Im Rahmen des Projekts wurde ein Modell entwickelt, das die systematische Analyse von Risikofaktoren für Softwareobsoleszenz ermöglichen soll. Das sogenannte LEUF-Schema (abgeleitet aus den aufeinanderfolgenden Schwerpunkten der Analyse, siehe Abbildung 1) stellt eine Synthese der Ergebnisse aus den folgenden Kapiteln dar. Es soll dabei helfen Risikofaktoren für Softwareobsoleszenz in Anwendung und Nutzung besser zu formalisieren und in einer einfachen Ereigniskette abzubilden. Das Modell ist eine Synthese aus den Vorschlägen von Sandborn (Sandborn 2007, siehe Kap. 2.2.1), dem ISO Standard zur Beschreibung von Softwarequalität (ISO 25000, siehe Kap. 2.2.2) und formuliert Kernfaktoren zur Vermeidung von Softwareobsoleszenz.

Das Schema zeigt eine Sequenz von aufeinander aufbauenden Voraussetzungen, von dem softwarebasierte Produkte abhängig sind. Diese werden kategorisiert als:

- LEGAL= Die Nutzenden müssen über die Nutzungsrechte der Software verfügen und diese muss rechtskonform einsetzbar sein (Datenschutz, Urheberrecht).
- EXECUTABLE = Die Software muss technisch mit der verbundenen Hardware ausführbar sein/ bleiben.
- USABLE = Nutzende müssen mit der Software angemessen umgehen können.
- FUNCTIONAL = Die Software muss ihren Zweck hinreichend erfüllen.

Abbildung 1: LEUF-Schema



Quelle: Eigene Darstellung, TU Berlin

Das LEUF-Schema zeigt die Interdependenz von verschiedenen Faktoren, die aus Sicht der Nutzenden für den Betrieb und Anwendung einer Software notwendig sind, wobei der Verlust eines Faktors zur absoluten Obsoleszenz des Produkts führen kann. Die Nutzung von Software ist beispielsweise überwiegend an eine Nutzungslizenz (legal) gebunden, aus der bestimmte Rechte und Pflichten des Bereitstellers der Software resultieren. Erlischt eine Lizenz oder wird vom Softwareanbieter nicht mehr verlängert, führt dies in der Praxis entweder dazu, dass der Hersteller die Nutzung der Software aus der Ferne unterbindet (software kill switches, digital

restrictions, soft bricks) bzw. verbundene Cloud-Services deaktiviert oder Nutzende die Weiternutzung der Software zwar ermöglicht, ohne jedoch für nötige Softwareupdates und -sicherheit in der Zukunft Sorge zu tragen. In beiden Fällen wird die Ausführung (executable) der Software behindert oder stellt zumindest ein Risiko für Nutzende dar. Darüber hinaus sind jedoch auch Fälle denkbar, bei den Nutzende zwar über eine gültige Nutzungslizenz (legal) für ihre Software verfügen und diese auch über ihre vorhandene Hardware ausführen (excetuble) können, in der Praxis aber Probleme bei der Benutzung haben (usability). Die Fähigkeiten und Kompetenzen der Nutzenden sowie das Nutzererlebnis können sehr unterschiedlich sein. Verändert sich beispielsweise in Folge eines Softwareupdates die Nutzeroberfläche einer Anwendung oder wird das Laden einer Anwendung als zu langwierig empfunden, so können auch diese subjektiven Wahrnehmungen dazu führen, dass Nutzende ihre Produkte vorzeitig ablegen. Das Beispiel zeigt, das in der Praxis nicht immer klar trennbar ist, welche Relevanz direkte software-induzierte Faktoren gegenüber indirekten Faktoren wie die Wahrnehmung der Nutzenden spielen (siehe Kap. 1.3.1, Tabelle 2).

Das LEUF-Schema ist ein vereinfachtes Modell von Wirkzusammenhängen, die aus der Perspektive von Nutzenden ein Risiko für Softwareobsoleszenz darstellen. Es erhebt keinen Anspruch auf Vollständigkeit und eine weitere empirische Operationalisierung und Validierung sind notwendig. Insbesondere der dynamische Charakter von Software und ein sich kontinuierlich änderndes Systemumfeld sowie Erwartungen der Nutzerenden stellen eine große Herausforderung für die Bestimmung und Messung von Softwarequalitäten dar.

2 Bestandsaufnahme und Stand der Forschung

In den letzten Jahren hat das Thema der Softwareobsoleszenz eine wachsende Aufmerksamkeit erfahren und wird in Deutschland insbesondere umweltpolitisch adressiert (siehe Kapitel 8). Im Jahr 2014 konnte im Rahmen des Green IT Projekts des Umweltbundesamts das Problem softwarebedingter Obsoleszenz bereits als umweltpolitische Herausforderung benannt werden (Gröger & Köhn 2015). Eine Studie zu Grüner Software im darauffolgenden Jahr wies zudem auf die software-bedingte Obsoleszenz von Hardware hin, konnte jedoch nicht umfangreicher über Ursachen und Wirkungen aufklären (Hilty et al. 2015). Mit der Veröffentlichung der Umweltpolitischen Digitalagenda des BMU (BMU 2020) und des Aktionsplans Kreislaufwirtschaft der Europäischen Kommission (EC 2020b) hat es das Thema endgültig auf die politische Agenda geschafft.

Wie im vorliegenden Bericht gezeigt werden kann, ist das Thema jedoch nicht nur aus Sicht der Umweltpolitik von Interesse, sondern auch aus Sicht der Nutzenden, Wirtschaft und Sicherheitspolitik. Die zunehmende Digitalisierung führt zu einem stärkeren Einsatz von vernetzten softwarebasierten Geräten in allen Lebensbereichen (Cyber-Physical Systems). Spezifische Software und elektronische Komponenten ermöglichen die Vernetzung mit dem Internet oder mit anderen Produkten, was zu einer höheren Technologieabhängigkeit führen kann. Softwareprodukte sind bisher jedoch im Vergleich zu anderen Produktbereichen wenig standardisiert und die Ursachen und Wirkungen von Softwareobsoleszenz kaum erforscht

Um den Stand der Forschung zur softwarebedingten Obsoleszenz zusammenzufassen, wurde eine umfangreiche Literatur- und Internetrecherche durchgeführt. Ziel war es, bestehende Untersuchungen und Konzepte zur Softwareobsoleszenz³ zusammenzufassen, um den Ausgangspunkt des Projekts besser zu definieren und eine klare Begriffsbestimmung und Operationalisierung vorzunehmen. Die Auswertung der Recherche erfolgte qualitativ und folgte dem Ziel, den aktuellen Erkenntnisstand übersichtlich darzulegen, weiterführende Hypothesen und Forschungsbedarfe aufzudecken sowie eine hinreichende Gegenstandsbeschreibung und Definition von Softwareobsoleszenz zu erarbeiten.

Die Analyse und Recherche stützte sich auf ein breites Feld an Medien (siehe Tabelle 3) und versuchte, bestehende Diskurse darzulegen sowie in verwandte Konzepte einzubetten.

Tabelle 3: Fokussierte Medien in der Literaturrecherche

Medien	Erkenntnisinteresse
Wissenschaftliche Publikation	Definition und Operationalisierung von Softwareobsoleszenz, Konzepte, Typologien, Klassifizierungen, Ursachen, Lösungen, Studien, verwandte Themenbereiche
Fachartikel und -magazine (bspw. Stiftung Warentest)	Wahrnehmung und Bewertung von Softwareobsoleszenz, Praxisbeispiele, Unterschiede zwischen Produktgruppen
Standards und Normen	Aktueller Stand der Normung, Operationalisierung von Softwareobsoleszenz, Problemauffassung und -beschreibung, Ermittlung von Schwerpunkten und Handlungsansätzen

³ Da in der vor allem medialen, aber auch zum Teil in Fachkreisen geführten Diskussion vor allem der Begriff Softwareobsoleszenz genutzt wird, wird dieser im vorliegenden Bericht ebenso häufig verwendet. Zu analytischen Zwecken ist jedoch der Begriff softwarebedingte Obsoleszenz zu bevorzugen.

Medien	Erkenntnisinteresse
Sonstige Publikationen	U.a. Erkenntnisstand aus Fachgesprächen, Standpunkte von europäischen und nationalen Ausschüssen, Diskussions- und Arbeitspapiere aus Arbeitskreisen (COG, Runder Tisch Reparatur) usw.
Fachforen im Internet	Problemauffassung und -beschreibung, Ermittlung von Praxisbeispielen, Schwerpunktthemen und -problembereiche
Sonstige Internetquellen	Suchmaschinenbasierte Stichwortsuche national und international, Gegenstandsbestimmung und -beschreibung

Das Suchraster basierte auf der Ermittlung von Schlagworten (Keywords), die in einem iterativen Prozess fortwährend ergänzt wurden. Die Suche wurde ergänzt durch einschlägige Publikationen, die bereits bekannt waren oder die über eine generische Suche über Suchmaschinen ermittelt wurden. Die Einzelquellen wurden einzeln gesichtet, hinsichtlich ihrer Relevanz bewertet (Bsp. Review von Abstracts) und in einer Datenbank systematisch erfasst und für die Detailauswertung verschlagwortet.

2.1 Hintergrund: Entwicklung der IKT-Industrie

Würden sich Technologien nicht weiterentwickeln, Lizenzen nicht auslaufen, Hardware keinem physischen Verschleiß unterliegen und die Anforderungen der Nutzenden sich nicht ändern – dann gäbe es wohl keine Softwareobsoleszenz. Das Gedankenspiel soll zeigen, dass die softwarebedingte Obsoleszenz im Kern den gleichen Wirkprinzipien unterworfen ist, wie sie bei anderen Arten der Obsoleszenz anzutreffen ist (Poppe & Longmuss 2019). Die Ursachen und Wirkungen von Softwareobsoleszenz sind kein neues Phänomen und finden sich entlang der gesamten Computer- und Softwarehistorie, die ihren Ausgangspunkt in den 1940er Jahren mit der Entwicklung der ersten nicht-zivilen Großrechner (Computer) nahm. Durch die sukzessive Entwicklung neuer Technologien und Erschließung neuer Anwendungsgebiete war die IKT-Industrie seit jeher geprägt von einer zunehmenden Diversifizierung von Hard- und Softwareprodukten. Die Diversifizierung ermöglicht neue vielfältige Anwendungsgebiete und war häufig Ausgangspunkt weiterer Innovationen. Obgleich Software keinem physisch-werkstofflichen Verschleiß unterliegt – kommt es auch zu beobachtbaren Alterungsprozessen bei Software, die gemeinhin auch als Softwareobsoleszenz oder Softwarealterung beschrieben werden kann.

Eine erste Erwähnung des Begriffs „software rot“ findet sich bereits 1983 in grauer Literatur (Hackers Dictionary), in Zeitschriften (Gerstl 2019/Graser 2015) und der Plattform Wikipedia wird das Problem unter Softwareerosion zusammengefasst, während in wissenschaftlicher Literatur bis heute unter den Begriffen Software-Qualität, Software Evolution und Software Aging bereits wesentliche Bestandteile von Softwareobsoleszenz definiert sind. Software-Qualität ist demnach eine Funktion, die abhängig ist von der Zuverlässigkeit, Fehlertoleranz, Sicherheit (System), Schutz (gegen außen), Verfügbarkeit, Testbarkeit und Wartbarkeit (Voas & Agrest 2004) der Software. Eine Verbindung dieser Herangehensweisen zur (sozial-ökologischen) Nachhaltigkeitsforschung ist zum aktuellen Stand nicht vorhanden.

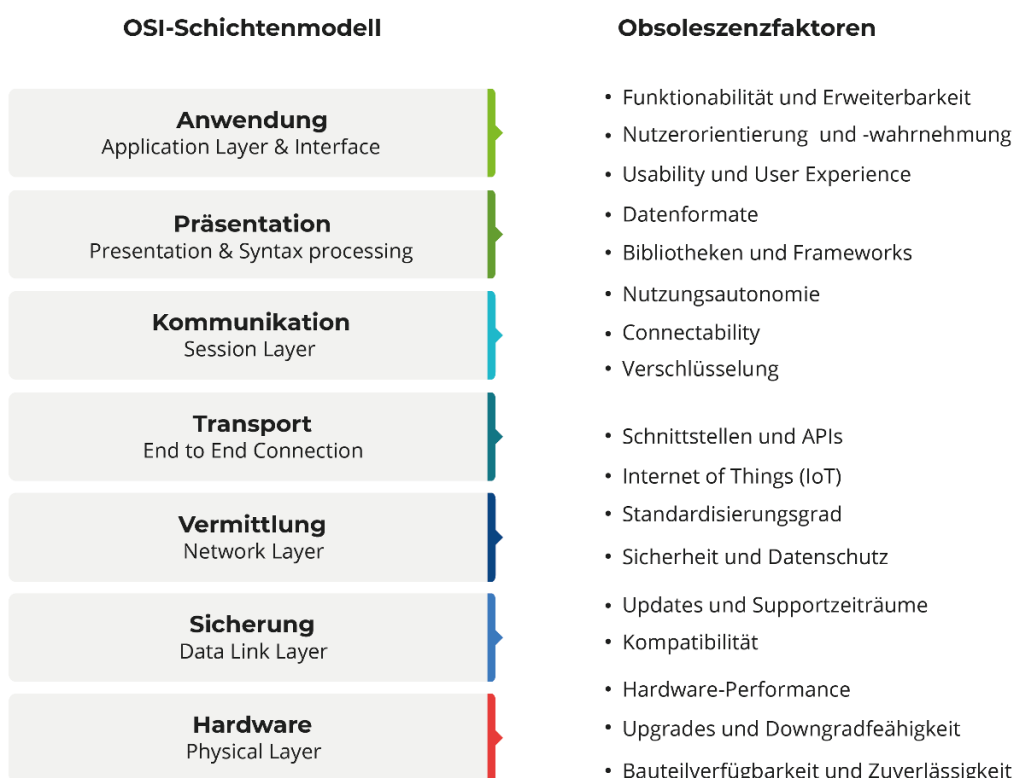
Der folgende Abschnitt dient dazu, auf wichtige strukturelle technologische Entwicklungen innerhalb der IKT-Industrie einzugehen, die eng mit dem Phänomen der Softwareobsoleszenz verknüpft sind. Hierzu gehören insbesondere die hohe Diversifizierung von Hard- und Softwareprodukten und der gleichzeitige Versuch, diese durch die Erhöhung von Kompatibilität, Interoperabilität und Vernetzung zu integrieren und standardisieren.

2.1.1 Modularisierung von Hard- und Software

Der Begriff der „Software“ geht zurück auf die Entwicklungen der Computerindustrie in den 1960er Jahren und bezeichnet den bis dahin geläufigeren Begriff der „Programme“ (Martin & Garcia-Swartz 2015: pos. 635/ Tukey 1958). Software bezeichnet ein Programm oder eine Menge an Programmen, die dazu dienen einen Computer zu steuern. Computer werden hierbei verstanden als Funktionseinheiten, die logische und arithmetische Berechnungen ohne menschliches Eingreifen durchführen können. In der Regel wird Software zwar als immateriell beschrieben, für deren Ausführung ist dennoch physisches Equipment notwendig, das in der Regel aus elektronischen Komponenten besteht und als „Hardware“ beschrieben wird (ISO/IEC-Norm 24765:2017).

Die ersten Computer realisierten Berechnungen über eine starre Kombination ihrer Hardwarekomponenten, sodass für verschiedene Anwendungen und Berechnungen diese im Vorfeld neu kombiniert und verschaltet werden mussten. Diese Rekonfiguration wurde später durch den Einsatz von Software in Form von Lochkarten automatisiert, in dem über die spezifische Kombination von Löchern auf einem Papier, die entsprechenden Schaltungen in der zugrundeliegenden Hardware realisiert werden konnten. Als Speichermedien für die entsprechende Software kamen später weitere Technologien wie Magnetbänder, Festplatten, Floppy Disks und Compact Disc hinzu. Zu Beginn der 1960er Jahre wurden die ersten Patente auf Softwareprogramme vergeben, was den eigenständigen Charakter von Software gegenüber Hardware unterstreicht. Bis in die Mitte der 60er Jahre waren die meisten Computer Großrechner in Unternehmen und öffentlichen Einrichtungen, die zwar mit verschiedener Software betrieben werden konnten, die Hardware und Software hierbei jedoch nicht kompatibel mit anderen Systemen war (Martin & Garcia-Swartz 2015: pos. 635/ Tukey 1958). Eine frühe Form der Softwareobsoleszenz bestand demnach darin, dass die Systeme nicht oder nur beschränkt an sich ändernde Anforderung angepasst werden konnten und zudem häufig nicht kompatibel mit neuen Technologien waren. Die US-Regierung war zu diesem Zeitpunkt einer der Großabnehmer der Rechner und drängte daraufhin auf eine stärkere Entkopplung von Hard- und Software, um den Wettbewerb zwischen verschiedenen Anbietern zu fördern und durch die Modularisierung eigene Kosten zu senken (Martin & Garcia-Swartz 2015: pos. 1203). Der Nutzen von Modularisierung ist dabei an die Bedingung geknüpft, dass die entkoppelten Hard- und Softwarekomponenten über kompatible Schnittstellen miteinander interagieren können.

Abbildung 2: Auswahl möglicher Obsoleszenzfaktoren entlang des OSI-Schichtenmodells



Quelle: Eigene Darstellung, TU Berlin

Eine entscheidende Entwicklung zur Förderung der Kompatibilität hat hierbei die Abstraktion von Hard- und Softwarearchitekturen mit Hilfe verschiedener Schichtenmodelle, Standards und Visualisierungsmethoden gespielt. Einige der Modelle, wie das OSI-Schichtenmodell (Open Systems Interconnection Model), werden heute noch eingesetzt, um verschiedene Spezifikationen und Funktionen innerhalb vernetzter Systeme zu beschreiben (siehe Abbildung 2). Schichtenmodelle und Standards sind abstrakt, sollen dem Menschen helfen, Komplexität zu reduzieren, und spiegeln daher nicht immer die zugrundeliegende Rechnerarchitektur wider (Lohmann 2015).

Die ab den 70er Jahren stärker einsetzende Modularisierung und Standardisierung sorgte für eine zunehmende vertikale Desintegration innerhalb der Systemarchitektur, sodass in den folgenden Jahrzehnten eine Vielzahl von neuen technischen Innovationen im Hardware- und Softwarebereich ermöglicht wurden. Eine vollständige Entkopplung und Modularisierung von Hard- und Softwarekomponenten, könnte prinzipiell die Voraussetzung dafür schaffen, jeder Form von Obsoleszenz effektiv zu begegnen – weil alle Komponenten flexibel austauschbar und funktional erweiterbar wären. In der Praxis ist dies bisher jedoch nur bis zu einem gewissen Grad möglich und geht einher mit wirtschaftlichen Zielkonflikten in den Geschäftsmodellen der Herstellenden, die bemüht sind, Kund*innen an ihre Produktsysteme und Plattformen zu binden („customer lock-in“) oder eigene Technologien zu schützen. Proprietäre (geschlossene) Systeme ermöglichen beispielsweise keinen Zugriff auf den Quellcode und sind von Dritten nur mit einer Lizenz des Herstellers nutz- und eingeschränkt erweiterbar, wohingegen freie Software und offene Standards mit entsprechenden Lizenzen frei nutz- und anpassbar ist (Albers 2015). Es handelt sich bei Softwareobsoleszenz demnach nicht nur um ein technisches Optimierungsproblem, sondern immer auch um ein politisch-wirtschaftliches Problem, weil es die Frage nach verpflichtenden Standards sowie Nutzungs- und Verwertungsrechten aufwirft.

2.1.2 Kompatibilität und Vernetzung

Aufgrund der zunehmenden Diversifizierung waren die Erhöhung der Kompatibilität und Standardisierung innerhalb der IKT immer wieder Ziel staatlicher Interventionen und Politik: sei es bei der Beseitigung von Herstellermonopolen durch die Entkopplung von Hard- und Software in den 1970er Jahren, der Schaffung eines einheitlichen Mobilfunkstandards ab den 1980er Jahren, fairen Wettbewerbsbedingungen bei Anwendungssoftware wie Internetbrowsern in den späten 90er Jahren sowie der Formulierung einheitlicher Daten- und Kommunikationsstandards. Obwohl Software immer wieder Gegenstand von Regulierungen war, ist sie bis heute im Vergleich zu anderen Produktbereichen wenig reguliert. Dabei hat gerade in den letzten zwei Jahrzehnten die Diversifizierung und der Bedarf an Vernetzung von softwarebasierten Produkten massiv zugenommen (Martin & Garcia-Swartz 2015:). Neue Obsoleszenzrisiken entstehen, welche die Sicherheit, Funktionalität und Kompatibilität von vorhandenen und künftigen Produktsystemen gefährden.

Internetbasierte Anwendungen sind heute hochgradig vernetzt und zum Teil abhängig von Softwaremodulen, die heute dezentral über das Internet bereitgestellt werden. Die weltweit größte Entwicklerplattform Github dient vielen Softwareentwickler*innen heute als „Repository“ (engl. Lager, Quelle), um einzelne Softwaremodule bis hin zu kompletten Softwarebibliotheken für sich und andere über eine Schnittstelle (application programming interface - API) zur Verfügung zu stellen. In ihrem jährlichen Bericht wertet die Plattform unter anderem Abhängigkeiten zwischen den verschiedenen Softwareprojekten aus. Paketmanager sind beispielsweise Programme, die verschiedene Softwarepakete, -module und Bibliotheken bestimmter Programmiersprachen und -umgebungen über eine einfache Schnittstelle Entwickler*innen bereitstellen und aktuell halten. Hier zeigt sich, dass insbesondere im Open Source Bereich die Arbeit von wenigen Entwickler*innen, Einfluss auf Millionen anderer Projekte nehmen kann. Im Fall des npm (Node Package Manager) Paketsoftware für die JavaScript-Laufzeitumgebungen, verantworten beispielsweise durchschnittlich 35 Softwareentwickler*innen die Weiterentwicklung der Paketsoftware, von der wiederum 3.5 Millionen andere Softwareprojekte auf Github unmittelbar abhängen (siehe Abbildung 3).

Abbildung 3: Durchschnittliche Mitwirkung und Abhängigkeiten pro Paketsprache auf Github.com im Jahr 2019

Average package contributors and dependencies by package language	Top 50 packages (for each package manager)	Avg. dependent projects	Avg. direct contributors
The top 50 open source packages in every language ecosystem (JavaScript, Python, Ruby, etc.) have an incredible number of dependent projects. For example, despite having an average of less than 40 direct contributors each, popular npm packages can be dependencies for millions of other repositories.*	Maven packages	167k	99
	pip packages	78k	204
	npm packages	3.5m	35
	NuGet packages	94k	109
	RubyGems packages	737k	146

Quelle: Bildschirmkopie, <https://octoverse.github.com/2019/>, 26.01.2022

Welche direkten Folgen diese Abhängigkeiten haben können, zeigt ein aktuelles Beispiel. Von dem fehlerhaften Update einer zweizeiligen JavaScript-Library des npm Paketmanagers waren Ende April 2020 zeitweise mehrere Millionen JavaScript-Projekte betroffen, darunter auch

Softwareframeworks wie Googles Angular.js, die wiederum eigene Abhängigkeiten für andere Projekte bereitstellen. Der Fehler konnte zwar schnell erkannt und behoben werden, zeigt jedoch die hochgradige Abhängigkeit von Softwareprodukten untereinander (Stoll 2020b).

Heute befindet sich Software in einer Vielzahl von Produkten und Komponenten, wo sie Steuerungs- und Kommunikationsaufgaben übernimmt. Häufig wird in diesem Kontext auch von „eingebetteten Systemen“ gesprochen. Eingebettete Systeme, die aus Elektronik und Software bestehen, gelten als ein Wachstumsmarkt der Deutschen Industrie und erweitern die Funktionalität von Fahrzeugen, Flugzeugen, medizinischen Geräten, Produktionsanlagen und von Haushaltsgeräten: „Schon heute arbeiten etwa 98 Prozent der Mikroprozessoren eingebettet, über Sensoren und Aktoren mit der Außenwelt verbunden. Zunehmend werden sie untereinander und in das Internet vernetzt. Die physikalische Welt verschmilzt mit der virtuellen Welt, dem Cyberspace. Es entstehen Cyber-Physical Systems (CPS), die Teil einer zukünftig global vernetzten Welt sind, in der Produkte, Geräte und Objekte mit eingebetteter Hardware und Software über Anwendungsgrenzen hinweg interagieren“ (acatech 2011; Geißberger & Broy 2012).

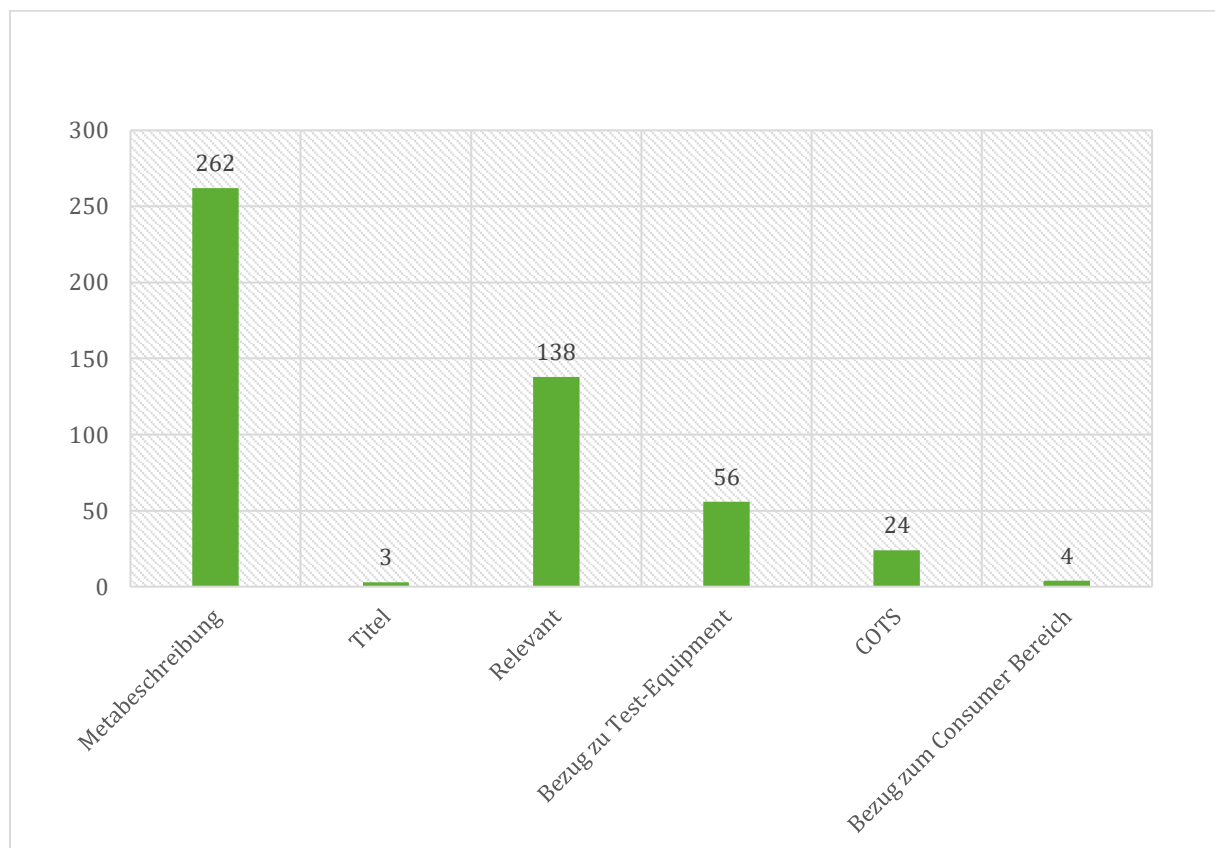
Ogleich es in den Jahrzehnten zuvor eine vertikale Desintegration in der Systemarchitektur gab, sind viele Hersteller heute Software- und Hardwarehersteller zugleich, wobei der Großteil den Fokus und die Wertschöpfung im Bereich der Software und den damit verbundenen Services sucht (Martin & Garcia-Swartz 2015). Dieser Prozess ging einher mit einer Marktentwicklung, die verstärkt ab Mitte der 1980er einsetzte und unter dem Begriff der „Consumerization“ im Bereich der Elektronik und IKT beschrieben wird. Die Möglichkeit der seriellen Produktion von Mikroelektronik und Halbleiterkomponenten lies die Gewinnmargen beim Verkauf von Hardware sinken und ermöglicht es heute, verhältnismäßig preiswert leistungsfähige Elektronikprodukte zu kaufen. Die Marktzyklen und Verfügbarkeit der Produkte sind jedoch kürzer und bedingen häufig die vorzeitige Obsoleszenz intakter Produkte (Ward & Sohns 2011/ Bartels & Poppe 2019).

Die zunehmende Digitalisierung aller Lebensbereiche und Ausstattung der Haushalte mit vernetzten Geräten ist ein unmittelbares Erbe dieser Entwicklung mit bereits absehbaren sozio-ökologischen Folgen. Eine Zunahme vernetzter Produkte in den privaten Haushalten, Industrie und nicht-zivilen Bereichen wird über kurz oder lang unvermeidbar zur Obsoleszenz und damit einem höheren Elektroschrott aufkommen führen. Mehr Produkte impliziert mehr Obsoleszenz (Poppe 2019). Es sei daher an dieser Stelle darauf hingewiesen, dass eine Minderung der software- und hardwarebedingten Obsoleszenz zwar zu einer Entschleunigung von Nutzungszyklen beitragen kann, jedoch mit hoher Wahrscheinlichkeit auch von Rebound-Effekten in Folge von Effizienz, Kompatibilitäts-, und Performancegewinnen begleitet wird, wie sie bisher üblich im Bereich der IKT sind (Santarius 2015). Statt einzig auf die relative Verbesserung in der Nutzungsdauer abzielen, sollte deshalb aus Sicht der Umweltpolitik geprüft werden, an welchen absoluten Einspar- und Vermeidungszielen sich die Verminderung von Softwareobsoleszenz messen lassen soll.

2.2 Forschung zum Thema softwarebedingte Obsoleszenz

Die Ermittlung des Stands der Forschung erfolgte in einem ersten Schritt systematisch über die Forschungsdatenbank IEEEExplore⁴. Die Forschungsdatenbank IEEEExplore bietet Zugang zu Artikeln, Konferenzberichten, technischen Standards und verwandten Materialien in den Bereichen Informatik, Elektrotechnik und Elektronik sowie verwandten Bereichen. Ziel der Suche ist es, einen ersten Überblick und Gradmesser zu erhalten, in welchen Bereichen und Kontext das Thema bereits unter dem Begriff der „Software Obsoleszenz“ behandelt wird.

Abbildung 4: Ergebnisse der Metasuche in IEEEExplore zu Software Obsoleszenz



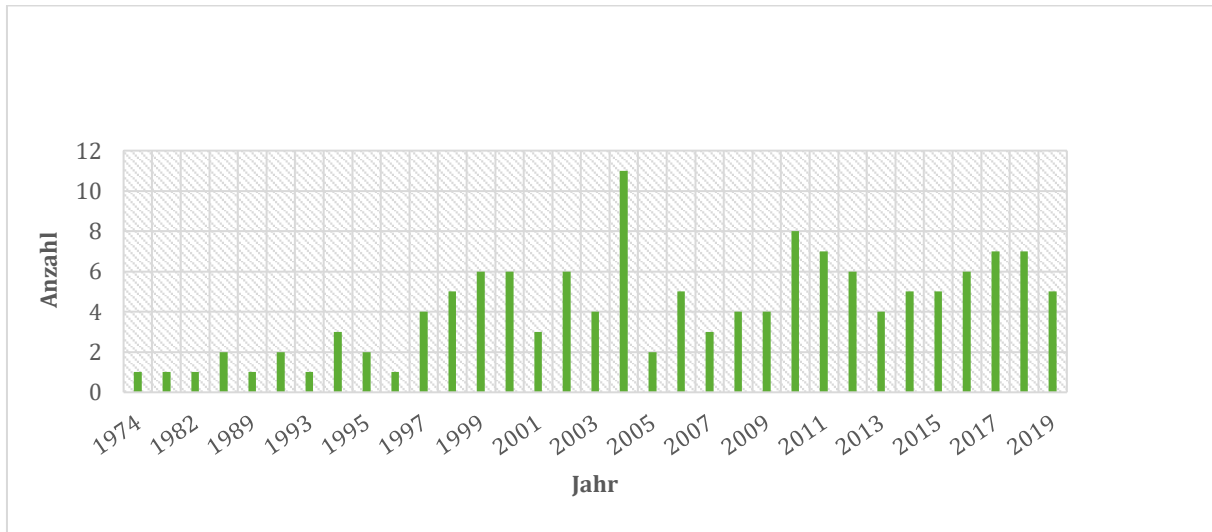
Quelle: Eigene Darstellung auf Basis der Metasuche in der Forschungsdatenbank: <https://ieeexplore.ieee.org/>, Stand 14.04.2020

In der initialen Suche (Volltext) konnten zunächst über 4.300 Ergebnisse ermittelt werden, die in einem weiteren Schritt auf Basis der Metaangaben (Titel, Abstract, Keywords, etc.) weiter kondensiert wurden. Abbildung 4 zeigt die Ergebnisse der Metasuche und Teilergebnisse aus der Metasuche. Der überwiegende Teil der Publikationen besteht dabei aus Konferenzbeiträgen. Es gibt lediglich drei Veröffentlichungen, die zum Zeitpunkt der Untersuchung im Titel direkt Bezug auf Softwareobsoleszenz nehmen. Alle Ergebnisse der Suche wurden einzeln gesichtet und auf ihre Relevanz bewertet (Relevant: 138). Bei der Auswertung ist aufgefallen, dass sich ein Großteil der relevanten Ergebnisse auf Obsoleszenzproblematiken im Bereich Test Equipment (Automatic Test Equipment, Test Program Sets) im Industrie- und Anlagenbereich bezieht. Es wird zwar im Rahmen von COTS (commercial off-the-shelf) auf Obsoleszenzproblematiken bei kommerziellen Produkten (Hard- und Software) hingewiesen (COTS: 24), direkte Bezüge zur kommerziellen Nutzung im Bereich Unterhaltungselektronik sind jedoch sehr selten (Bezug zu

⁴ URL <https://ieeexplore.ieee.org/Xplore/home.jsp>

Consumer: 4). Wie bereits unter Kapitel 1.3.1 ausgeführt wurde, werden Obsoleszenzproblematiken von und durch Software in Fachdiskursen auch in anderen Kontexten adressiert, sodass nur ein Abgleich und weiterführende Recherchen hier Aufschluss über die tatsächliche Relevanz geben kann.

Abbildung 5: Relevante Veröffentlichungen pro Jahr im IEEEExplore



Quelle: Eigene Darstellung auf Basis der Metasuche in der Forschungsdatenbank: <https://ieeexplore.ieee.org/>, Stand 14.04.2020

Abbildung 5 zeigt die Anzahl relevanter Veröffentlichung auf Basis der Suche in der IEEEExplore Forschungsdatenbank. Wie zu erkennen ist, gibt es seit Mitte der 90er Jahre eine leichte Zunahme der Veröffentlichungen. Der Ausschlag im Jahr 2004 steht im Zusammenhang mit der AUTOTESTCON Konferenz, die sich im Bereich automatischer Testsysteme für technische Anwendungen im Militär- und öffentlichen Sektor spezialisiert und in dem betreffenden Jahr unter anderem das Thema der Migration von alten auf neue Testsysteme hatte.

Die geringe Ergebnisauswahl stützt die These, dass Softwareobsoleszenz bisher nur wenig operationalisiert und als Fachterminus verbreitet ist. Dennoch sollen die Ergebnisse im Folgenden unter Bezugnahme auf weitere Fachquellen ausgewertet werden, um Besonderheiten, Schwerpunkte, Methoden und erste Erkenntnisse in diesem Bereich zu identifizieren.

2.2.1 Obsoleszenz im Industrie- und Anlagenbereich

Obwohl der Schwerpunkt der vorliegenden Studie nicht im industriellen, öffentlichen und militärischen Bereich liegt, soll im Folgenden auf Entwicklungen eingegangen werden, die in einem engen Zusammenhang mit der Anwendung von Consumer Electronics (commercial off-the-shelf, kurz „COTS“) in langlebigen Industrie- und Anlagengütern stehen. Hinzu kommt, dass der Großteil der Forschungsarbeiten zu Softwareobsoleszenz im Kontext von industriellen Anwendungen erfolgt (siehe Abbildung 4). Industriegüter zeichnen sich in der Regel durch lange Einsatzzyklen von bis zu 20+ Jahren aus und unterliegen daher immer Obsoleszenz (DIN 62402:2017-09). Zum Erhalt ihrer Systeme haben Industrie und Anlagenbetreiber ein Problembewusstsein für Obsoleszenz und Vermeidungsstrategien entwickelt, die auch Erkenntnisse für den kommerziellen Bereich bieten.

Die einschlägigsten Arbeiten zum Thema Softwareobsoleszenz im industriellen Kontext liefert Peter Sandborn, der mit seinen Veröffentlichungen maßgeblich zur Systematisierung von

Obsoleszenzmanagement Methoden und insbesondere Forecasting Modellen beigetragen hat (Bartels et al. 2012). Sandborn stellt 2007 in einer Veröffentlichung fest, dass der überwiegende Teil von Methoden, Datenbanken und Tools im Bereich des Obsoleszenzmanagement sich zu einseitig auf Hardware-Lebenszyklen fokussiert, wobei die Gesamtkosten der Betreuung und Anpassung von Software über den gesamten Lebenszyklus häufig die der Hardware übersteigt. Nach Sandborn wird Softwareobsoleszenz bis dato nicht gut verstanden und in der Praxis schlecht bewältigt. Der Einfluss im Kontext von Obsoleszenzmanagement wird zudem weitestgehend ignoriert (Sandborn 2007). Darüber hinaus präsentiert er eine Systematik zu den Ursachen von Softwareobsoleszenz, speziell in Bezug auf kommerzielle Software (COTS), wie in der folgenden Tabelle ersichtlich ist:

Tabelle 4: Ursachen von Softwareobsoleszenz (n. Sandborn 2007)

Ebene	Beschreibung
Funktionale Obsoleszenz	Hardware-, Anforderungs- oder andere Softwareänderungen am System, vermindern die Funktionalität der Software (einschließlich durch Hardwareobsoleszenz ausgelöste Softwareobsoleszenz; und Software, die Software veraltet).
Technologische Obsoleszenz	Der Verkauf und/oder Support für Software läuft aus, indem: <ul style="list-style-type: none"> • der ursprüngliche Lieferant die Software nicht mehr neu verkauft (end of sale), • es nicht möglich ist die Lizenz zu erweitern oder erneuern (rechtlich nicht beschaffbar), • die Software-Wartung ausläuft, indem der ursprüngliche Lieferant und/oder Dritte die Software nicht mehr supporten (end of support).
Logistische Obsoleszenz	Die Veralterung (Obsoleszenz), Formatierung oder Degradierung digitaler Medien beschränkt oder verhindert den Zugang zur Software.

Ergänzend zur Typologie von Sandborn werden zudem vereinzelt Obsoleszenzrisiken in Folge einer alternden Belegschaft sowie der Verlust von Mitarbeiterkompetenzen und Anwenderwissen („Skills“, „Orgware“) benannt. Gerade bei älteren Systemen und Programmiersprachen kann dies der Fall sein (Rojo et al. 2010; Trimmer et al 1998, Sandborn & Prabhakar 2015). In den USA werden derzeit beispielsweise dringend Entwickler*innen für die 60 Jahre alte Programmiersprache Cobol gesucht, da die Datenverarbeitungssysteme einiger Ministerien, Banken und Versicherungen zum Teil darauf beruhen. Die Sprache gilt seit den 1980er Jahren jedoch als veraltet („Legacy Language“) und es gibt keine aktive Entwickler-Community mehr (Stoll 2020a/ Reuters 2020)⁵.

Das Problem der Softwareobsoleszenz ist nicht neu und ist seit jeher Teil von computergestützten Systemen. Neu ist hingegen eine Marktentwicklung, die verstärkt ab Mitte der 1980er Jahre einsetzt und unter dem Begriff der „Consumerization“ im Bereich der Elektronik und IKT beschrieben wird. Durch die Verlagerung des Leitmarktes für Mikroelektronik und Halbleiterkomponenten auf den kurzlebigen Consumersektor sahen sich Betreiber langlebiger Anlagen mit neuen Möglichkeiten aber auch Herausforderung konfrontiert. Komponenten (Hard- und Software) aus der kommerziellen Serienfertigung

⁵ Viele ältere Programmiersprachen sind dagegen heute noch erfolgreich im Einsatz. Die Programmiersprache „Fortran“ wird heute noch bei Simulationen auf der Suche nach Exoplaneten eingesetzt. Die Programmiersprache „Python“ wurde Anfang der 1990er entwickelt, fristete lange Zeit ein Nischendasein und verzeichnet heute ein wachsende Entwickler*innen Community im Bereich der Data Science.

(COTS) sind zwar günstiger, weisen aber in der Regel einen kürzeren Lebens- und Marktzyklus auf (Ward & Sohns 2011; Bartels & Poppe 2019).

Ein Sektor, der sich frühzeitig mit den neuen Möglichkeiten und Risiken beim Einsatz von COTS Software auseinandersetzt und bis heute einen Großteil der nennenswerten Veröffentlichungen und Fallstudien bereitstellt, findet sich im Bereich des Testens von technischen Systemen sowie Messsystemen speziell im Flug-, Militär, Weltraum- und Kraftwerksbereich. Flugzeuge, militärische Systeme aber auch Fertigungsanlagen in der Industrie sind mit einer Reihe von Systemen ausgestattet, die Computer gesteuert einzelne Komponenten oder ganze Systeme auf ihre Funktionsfähigkeit testen sollen. Mess- und Testinstrumente sind dabei häufig länger im Einsatz als die zu testenden Systeme und übersteigen zum Teil die Verfügbarkeitsdauer von Lieferanten und Techniker*innen (Toku 2013/ Lowenstein et al. 2015/ Tacha et al. 2010 / Neal 2004) – entsprechend langlebig wurde gebaut. In den 1990er Jahren zeichnete sich allmählich eine Trendwende ab, da durch den Zugriff auf leistungsfähige Mikroelektronik aus der kommerziellen Serienfertigung (COTS) es nun möglich war, kostengünstig aber dennoch „up-to-date“ die bestehenden Systeme zu erweitern oder zu ersetzen (Moorhead 1993). Im Einsatz von COTS Hard- und Software werden dabei verschiedene Vor- und Nachteile gesehen. Zum einen können veraltete Testsysteme durch die Erweiterungen von neuen Softwarefunktionalitäten (software injection) aktualisiert werden (Hardenburg 1999/ Watts 2016/ O'Donnell & Krayewsky 2006), zum anderen können neue Systeme durch einen modularen Aufbau, der Trennung von Soft- und Hardware und Virtualisierung langfristig dynamisch neuen Anforderungen angepasst werden (Heagney & Walker 2018/ Dano 2019/ Dzuik 2014). Die Nachteile sind jedoch, dass COTS Hard- und Software häufig nicht gut dokumentiert sind (Sparr et al. 2018, Morris 2000, Gardner 2016), bereits im Voraus eine durchdachte Systemarchitektur erfordern (Wubbena 2007/ Harnack 2012/ Robinson et al. 2015), und die Produktzyklen und Supportzeiträume deutlich verkürzt sind (Burrus et al. 2012).

Es zeigt sich in diesem Fall der komplementäre Charakter beim Einsatz von COTS: Konsumerelektronik und -software gilt bei langlebigen Produktsystemen als Lösungsbringer und Problemverursacher gleichermaßen. Es kommt zu einer Intensivierung und Beschleunigung der Obsoleszenzproblematik (Ward & Sohns 2011), der nur durch geeignete Design- und Managementstrategien begegnet werden kann (Bartels & Poppe 2019). Die Langlebigkeit eines Systems wird hier vor allem durch den proaktiven Austausch von Komponenten (Hard- und Software) und nicht durch die möglichst dauerhafte Auslegung einzelner Teile erzielt. Ein primäres Ziel industrieller und öffentlicher Anlagenbetreiber ist es deshalb, Obsoleszenzrisiken frühzeitig zu erkennen und präventive Maßnahmen zu ergreifen. Hierzu zählen spezielle Bewertungs- und Scoringmodelle bei der Beschaffung von COTS Software, die insbesondere nicht-funktionale Qualitätsmerkmale (non-functional constraints) in die Betrachtung einbeziehen (Wu et al. 2006, Morris 2010) sowie diverse Forecasting Methoden. Eine erste umfangreiche Übersicht hierzu bietet die systematische Literaturanalyse von Bowlds et al. aus dem Jahr 2018 (Bowlds et al. 2018).

Neben Versuchen der Operationalisierung und Sichtbarmachung von softwarebedingten Obsoleszenzrisiken werden zudem Vorschläge zu selbstadaptiver Software gemacht (Torres 2013), eine Anforderung, die in Studien des Umweltbundesamts und dem Blauen Engel auch unter dem Begriff der „bedarfsgerechten Steuerung“ angeführt wird (Hilty et al. 2017/ Blauer Engel 2020). Hierunter versteht sich, dass Software in der Anwendung und bei Updates nicht zu einer Performancedegradierung der verwendeten Hardware führt – also Software in ihrer Funktionalität auf die jeweiligen Hardwarekapazitäten optimal abgestimmt ist. Dies kann sowohl statisch erfolgen, bei Auslegung der Software auf ein bestimmtes System, aber auch

dynamisch in einem laufenden System. Bei letzterem werden Lasten auf verschiedene Systeme oder Prozessoren verteilt („Load Balancing“) oder andere Funktionen abgeschaltet, um Systemkapazitäten freizugeben. Obwohl die dynamische Verteilung von Lasten in vielen computergestützten Produktsystemen auch im Consumerbereich Standard ist, stellt sich im Speziellen die Frage, welchen Funktionen im System Vorrang gegeben wird und wer darüber entscheiden darf. Der Smartphone Hersteller Samsung informiert seine Nutzer beispielsweise darüber, dass Apps zur Leistungsoptimierung paradoxerweise dazu führen können, „dass dein Smartphone/Tablet langsamer wird oder gar nicht reagiert“ (Samsung 2020). Die Problematik ist vergleichbar mit dem beim autonomen Fahren und anderen offenen Systemen: die Software braucht Vorgaben, welche Funktionalitäten Vorrang vor anderen haben sollen.

Zusammenfassend konnte gezeigt werden, dass der Einsatz von COTS-Software im Industrie- und Anlagenbereich ein zweiseitiges Schwert darstellt: Zum einen konnte ab Mitte der 90er Jahre veraltete Hardware durch die „Software injection“ zu neuem Leben und Performance erweckt werden, zum anderen resultieren hieraus aber neue Obsoleszenzrisiken, weil COTS häufig schlechter dokumentiert sind und nur vergleichsweise kurze Supportzeiträume bieten. Obwohl sich die Branche frühzeitig mit den Risiken auseinandersetzte und eigene Lösungen wie dem Obsoleszenzmanagement entwickelte sowie neue Designprinzipien erprobte, ist auch hier das Wissen um softwarebedingte Obsoleszenz bisher gering und wenig formalisiert.

2.2.2 Qualitätsstandards und Normen

Software als immaterielles Gut und die damit verbundenen direkten und indirekten Wirkungen sind bisher nur schwer mess- und operationalisierbar (Hilty et al. 2017). Die Messbarkeit und Operationalisierung sind jedoch eine wesentliche Voraussetzung für die Bewertung von Eigenschaften (Qualitäten), die mit der Software verbunden sind. Bereits in den 1990er Jahren wurde hierfür von einem internationalen Gremium ein Standard zur Formulierung von Softwaremetriken erarbeitet, der jedoch lediglich als Prozessbeschreibung gedacht ist und keine konkreten Zielkennzahlen oder Bewertungsmaßstäbe formuliert (IEEE 1998).

Seit 2005 gibt es zudem die internationale Normenreihe der ISO/IEC 25000, welche spezifische Anforderung und Qualitätskriterien für die Bewertung von Softwareprodukten und deren Entwicklung beschreibt (ISO/IEC 25000:2014)⁶. Der ISO/IEC 25000 Standard definiert Softwarequalität auf Basis von funktionalen Aspekten (correctness, consistency, dependability, usability) und nicht-funktionalen Aspekten (performance, maintainability, security aspects). In der Differenzierung wird deutlich, dass Qualität hier als Gesamtheit von direkten und indirekten Eigenschaften eines Software-Produkts verstanden werden muss. Dabei geht es nicht nur darum, was ein System tut bzw. tun soll, sondern auch wie gut es die Funktionen erledigen soll (Balzer 2009: 465). Der Standard selbst verzichtet bewusst auf feste Zielvorgaben- und Erreichungsgrade und formuliert lediglich eine Auswahl an Kriterien und Prinzipien zur Festlegung von Anforderungen an das Softwaresystem, worunter sich auch eine Vielzahl an Obsoleszenzfaktoren findet (siehe Abbildung 6). Die Anforderungen selbst werden vom Hersteller, Auftraggeber und Stakeholder mehr oder weniger standardkonform und verbindlich festgelegt. Es ist anzunehmen, dass insbesondere größere Unternehmen sich an diesem Standard orientieren bzw. Elemente davon in ihrer Projektierung und Auftragsvergabe beachten. Wie bei vielen Standard erfolgt die Anwendung jedoch auf freiwilliger Basis, kann aber dennoch Bindungskraft als Gegenstand von Lieferverträgen und Teil der Lieferantenanforderungen entfalten.

⁶ ISO 25010x Reihe ersetzt seit 2005 die Norm ISO/IEC 9126, welche ausschließlich Kriterien zur Qualität der Software als Produkt enthält und nicht auf die Qualität des Softwareentwicklungsprozesses abzielt.

Abbildung 6: Qualitätskriterien der ISO/IEC 25010



Quelle: iso25000.com

Der Standard selbst macht keine expliziten Aussagen zu softwarebedingter Obsoleszenz, benennt in seinen Kriterien (siehe Abbildung 6) aber bereits relevante Kernaspekte wie Benutzerfreundlichkeit, Zuverlässigkeit und Wartbarkeit (ISO/IEC 25010). Er bietet damit die Möglichkeit konkrete Vorgaben zur erwartbaren Laufzeit, Interoperabilität, Abwärtskompatibilität und anderen Obsoleszenzfaktoren an das Softwaresystem zu formulieren. Weitere empirische Untersuchungen könnten zeigen, inwieweit der Standard in der Praxis überhaupt angewandt wird und in welcher Form hier bereits konkrete Anforderungen an Obsoleszenzfaktoren im Sinne eines impliziten Obsoleszenzmanagement⁷ gestellt werden. Der Standard gilt als internationale Referenz für die qualitative Bewertung von Softwaresystemen, es wäre deshalb zu empfehlen, dass Obsoleszenzfaktoren, wie beispielsweise Mindestlaufzeiten, Updatezyklen und geplante Supportzeiträume als Teilkriterien in Zukunft eine explizitere Erwähnung in dem Standard finden.

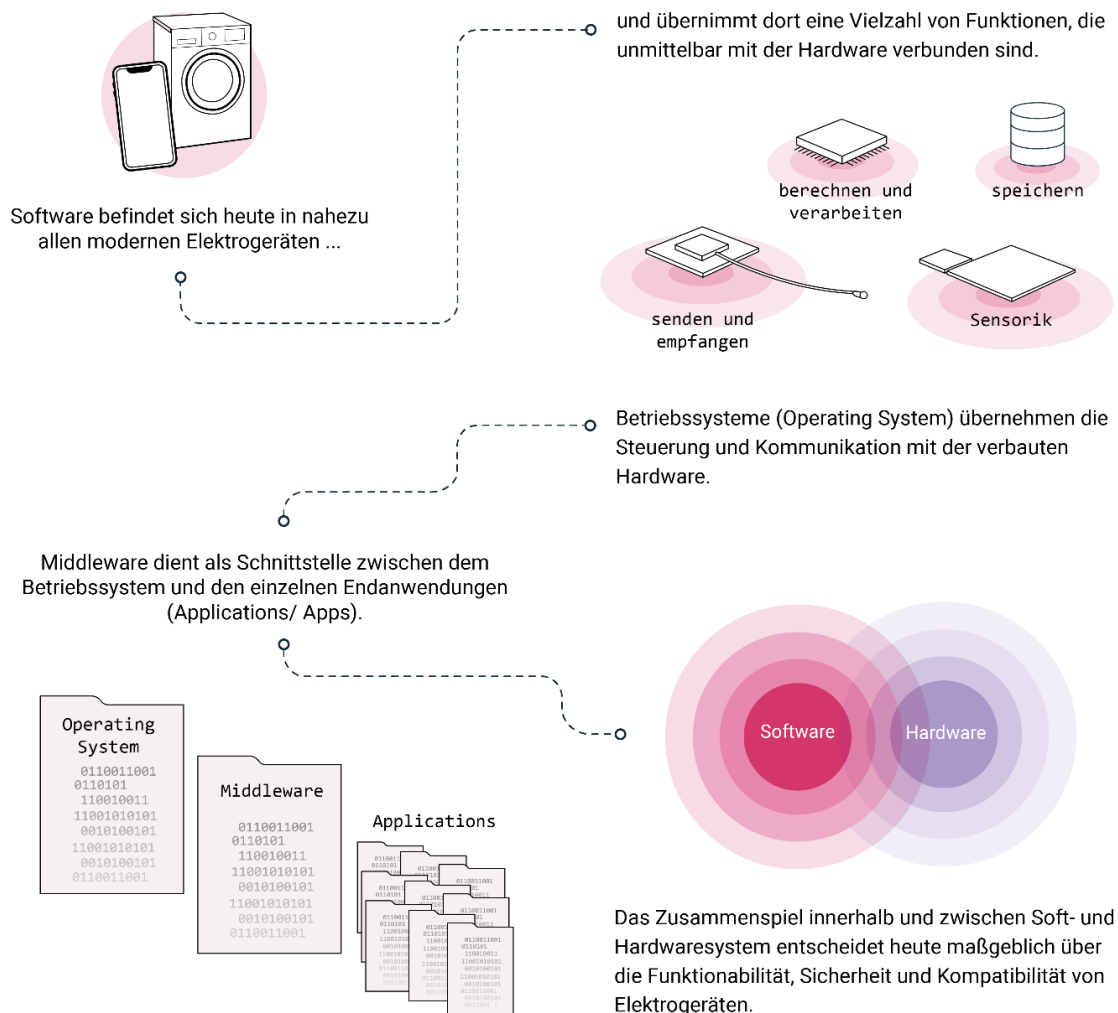
Mit dem Umweltzeichen Blauer Engel für „Ressourcen- und energieeffiziente Softwareprodukte“ (DE-ZU 215) können erstmals seit dem Jahr 2020 Produkte ausgezeichnet werden, die durch einen sparsamen Einsatz von Hardwareressourcen einen längeren Einsatz der Hardware-Nutzungsdauer ermöglichen sollen. Das Umweltzeichen richtet sich zunächst an endnutzerbezogene Anwendungssoftware wie Programme zur Bildbearbeitung, Textverarbeitung, Tabellenkalkulationen, E-Mail-Programme oder Webbrowser. Gefordert sind unter anderem die Gewährleistung der Abwärtskompatibilität, 5 Jahre verpflichtende Sicherheitsupdates nach Bereitstellungsende und die Vermeidung der Hardware-Degradation in Folge von Softwareupdates (Blauer Engel 2020). Das Umweltzeichen wurde vom Umweltbundesamt in Zusammenarbeit mit externen Expert*innen erarbeitet und basiert teilweise auf einer Vorstudie zur Ermittlung eines Kriterienkatalogs für nachhaltige Software (Hilty et al. 2017).

2.2.3 Relevanz von Software bei der Lebensdauer von Produkten

Software spielt heute in nahezu allen modernen Elektrogeräten eine maßgebliche Rolle und nimmt einen immer stärkeren Einfluss auf die potenziellen Nutzungsdauern. Mit der Zunahme software-betriebener Geräte (embedded systems) ist die Zuverlässigkeit und Funktionalität nicht mehr nur eine Frage der Hardware (siehe Abbildung 7).

⁷ „Obsoleszenzmanagement umfasst die Analyse von Obsoleszenzrisiken entlang der Wertschöpfungs- und Lieferkette, die Anwendung optimaler Lösungsstrategien zur Vermeidung und Reduzierung von Produktions- oder Dienstleistungsausfällen sowie den Informationsaustausch zu relevanten Lebenszyklusdaten von Produkten innerhalb der Lieferkette und gegenüber externen Stakeholdern.“ (Bartels & Poppe 2019: 130)

Abbildung 7: Zusammenspiel von Soft- und Hardware in modernen Elektrogeräten



Quelle: Eigene Darstellung, TU Berlin

Software übernimmt heute eine Vielzahl an zentrale Steuerungsaufgaben in Haushalts- und Unterhaltungselektronik und stellt maßgebliche Funktionalitäten bereit. Wie groß jedoch der tatsächliche Einfluss von Software auf die Produktlebens- und Nutzungsdauer von Produkten ist, war bisher nicht Gegenstand größerer Untersuchungen und wird in den folgenden Kapiteln anhand von Fallstudien (siehe Kapitel 3 und 5) und einer Analyse von Nutzer*innen-Erfahrungen (siehe Kapitel 6) näher beleuchtet.

Ein Großteil der physischen Hardware in elektronischen Geräten wäre heute ohne Software nicht nutzbar. Dass mit dem zunehmenden Einfluss der Digitalisierung und Marktentwicklung bei vernetzten Geräten (siehe Kap. 4) auch die Abhängigkeit von Software steigt, wurde bereits politisch erkannt (siehe Kap. 8). Seit dem Jahr 2021 setzt die Ökodesign-Richtlinie ((EU) 2009/125/EG) zudem Mindestanforderungen an die Verfügbarkeitsdauer von Software und Firmware für einige Produkte. Hierzu zählen unter anderem Server, Displays, Spülmaschinen oder Waschmaschinen. Andere Produktgruppen sollen mit der Überarbeitung weiterer Ökodesign-Richtlinien folgen.

2.3 Zusammenfassung und Fazit

Softwareobsoleszenz ist kein neues Phänomen und findet sich entlang der gesamten Computer- und Softwarehistorie, die ihren Ausgangspunkt in den 1940er Jahren mit der ersten Entwicklung nicht-ziviler Großrechner (Computer) nahm. Die Recherche hat bisher offengelegt, dass das Thema der softwarebedingten Obsoleszenz innerhalb der Fachwelt in unterschiedlichen Diskursen bereits adressiert wird, das Problem an sich jedoch nur wenig wissenschaftlich untersucht und formalisiert ist. Softwareobsoleszenz kann im Allgemeinen als Verlust von funktionalen (direkten) und nicht-funktionalen (indirekten) Eigenschaften von Software und die mit ihr verbundenen Produktsysteme beschrieben werden. Im Allgemeinen werden die verschiedenen Erscheinungsformen oder Teilaspekte von softwarebedingter Obsoleszenz bisher unter verschiedenen Begriffen wie dem „software rot“, „software erosion“, „software aging“ behandelt. Unabhängig vom Kontext und der Wirkungsweise teilen alle Konzepte dabei die Annahme, dass die langfristige Funktionsfähigkeit von Software von bestimmten Qualitätsmerkmalen abhängig gemacht werden kann, die sich jedoch im Zeitverlauf durch sich ändernde interne und externe Anforderungen verändern und degradieren können. Die IKT-Industrie ist seither geprägt von kurzen Markzyklen, einem hohen Innovationstempo und einer zunehmenden Diversifizierung von Soft- und Hardwaretechnologien, sodass eine Verallgemeinerung, Generalisierung und Standardisierung von Software-Qualitäten bisher sehr schwerfallen. Im Rahmen der Bestandsaufnahme und Recherche konnten zwar bereits zentrale Qualitätsstandards und Normen wie die ISO/IEC 2500x Serie zur Bestimmung der Softwareproduktqualität identifiziert werden, deren tatsächliche Anwendung und Umsetzung in der Praxis bleiben aber fraglich und erfordern tiefergehende empirische Untersuchungen. Die Aufnahme von Mindestanforderungen und Kriterien für die Beschaffenheit von Software bei einzelnen Ökodesign-Richtlinien ((EU) 2009/125/EG) seit dem Jahr 2021 sowie die Ergänzung der Warenkaufrichtlinie ((EU) 2019/771) um Informationspflichten für digitale Elemente zeigen jedoch den wachsenden Bedarf für die stärkere wissenschaftliche Formalisierung der nutzungs- und lebensdauerrelevanten Eigenschaften von Software.

3 Wirkungszusammenhänge von Soft- und Hardware

In diesem Arbeitspaket soll Software in seiner zugrunde liegenden Softwarearchitektur aufgeschlüsselt und zusammen mit der Hardware in einen Systemkontext gestellt werden. Zwischen den einzelnen Elementen („Schichten“) dieses Systems bestehen Abhängigkeiten, aus denen Obsoleszenzrisiken entstehen, also Risiken, die die Funktionalität des Gesamtsystems gefährden. Dafür ziehen wir zunächst ein weit verbreitetes Modell für Softwarearchitektur heran, das OSI Schichtenmodell (Open Systems Interconnection Model), und entwickeln es dahingehend weiter, dass weitere Abhängigkeiten und Systemzusammenhänge dargestellt werden können. Ausgehend vom OSI Schichtenmodell untersuchen wir den weiteren Systemzusammenhang, in den Produkte eingebettet sind, die Software enthalten. Neben den funktionalen Abhängigkeiten des OSI Modells (siehe Kapitel 3.1) werden auch Geschäftspraktiken und das Wirkungsgefüge der Akteure des erweiterten Systemzusammenhangs (Soft- und Hardwareentwickler*innen, Vertriebsplattformen, Anwender*innen) beleuchtet (siehe Kapitel 3.3).

Hauptsächlich soll der Frage nachgegangen werden, welche Schwachstellen sich innerhalb der einzelnen Schichten und insbesondere zwischen diesen ergeben und welche Einflussmöglichkeiten ein Softwareunternehmen bzw. ein Hersteller von Hardware hat, diese Schwachstellen zu umgehen.

Dieses differenzierte Wirkungsgefüge wird auf drei Beispiele angewandt, um die jeweils größten Obsoleszenzrisiken und daran beteiligten Akteure herauszuarbeiten. Aus diesen Erkenntnissen werden politische Handlungsempfehlungen abgeleitet, die einer softwarebedingten Obsoleszenz entgegenwirken.

3.1 Software Schichtenmodell

OSI steht für Open System Interconnection (Offenes System für Kommunikationsverbindungen) und wurde von der ISO (International Organization for Standardization) als Grundlage für die Bildung von offenen Kommunikationsstandards entworfen. Das OSI-Schichtenmodell besteht aus sieben Softwareschichten. Dabei definiert jede Schicht bestimmte Aufgaben und Funktionen für die Kommunikation zwischen zwei IT-Systemen über ein Netzwerk sowie Anforderungen an die sichere und zuverlässige Datenübertragung. Den einzelnen Schichten sind standardisierte Kommunikationsprotokolle zugeordnet, die eine Menge von Regeln enthalten und somit die Kommunikation an den Schnittstellen zwischen den Schichten ermöglichen. Tabelle 5 gibt eine Übersicht über die sieben Schichten des OSI Modells. Die Schichten 1 bis 4 sind transportorientierte Schichten, d.h. diese bilden die Basis der Datenübertragung. Die Schichten 5 bis 7 sind anwendungsorientierte Schichten.

Protokolle sind eine Sammlung von Regeln zur Kommunikation auf einer bestimmten Schicht. Weil manche Protokolle für bestimmte Anwendungen entwickelt wurden, kommt es auch vor, dass sich Protokolle über mehrere Schichten erstrecken und mehrere Aufgaben abdecken.

Tabelle 5: Übersicht der OSI Schichten

OSI Schicht	Funktionalität	Protokolle
Bitübertragung (Physical Layer)	Interpretation physikalischer Größen eines Mediums als Information	z.B. Ethernet
Sicherung (Data Link Layer)	Bündeln der Bits in Paketen	z.B. WLAN
Vermittlung (Network Layer)	Routing	z.B. IP

OSI Schicht	Funktionalität	Protokolle
Transport (Transport Layer)	Einsammeln von Paketen	z.B. TCP
Kommunikation (Session Layer)	Bündeln von Datenpaketen zu Sessions	z.B. PAP
Darstellung (Presentation Layer)	Interpretation und Struktur der Daten	z.B. ASCII
Anwendung (Application Layer)	Bereitstellen von Funktionalitäten	z.B. HTTP

Quelle: eigene Darstellung, Öko-Institut

3.1.1 Bitübertragungsschicht / Physical Layer

Diese unterste Softwareschicht interpretiert Änderungen physikalischer Größen als Informationseinheit (Bits), z.B. die Magnetisierung oder elektrische Spannung. Die Hardware und das Übertragungsmedium (z.B. ein Kupferkabel oder eine Antenne) sind nicht Bestandteil dieser Schicht. Die Protokolle dieser Schicht unterscheiden sich nach dem eingesetzten Übertragungsmedium und -verfahren.

3.1.2 Sicherungsschicht / Data Link Layer

Die Sicherungsschicht sorgt für eine zuverlässige und funktionierende Verbindung zwischen Endgerät und Übertragungsmedium. Dazu werden Datenpakete mit einer individuellen Hardwareadresse versehen. Ein Datenpaket verfügt über eine definierte Länge und Form, daher kann es auf Vollständigkeit und Brauchbarkeit geprüft werden. Zur Vermeidung von Übertragungsfehlern und Datenverlust enthält diese Schicht Funktionen zur Fehlererkennung, Fehlerbehebung und Datenflusskontrolle.

In dieser Schicht operiert z.B. IEEE 802.11 (die Normen des WLAN, der Marketingbegriff lautet Wi-Fi). Das Ethernet-Protokoll legt sowohl die physische Schicht (Layer 1) als auch die Data-Link-Layer fest.

3.1.3 Vermittlungsschicht / Network Layer

Auf dieser Schicht erfolgt erstmals die logische Adressierung der Endgeräte. Diese ist, im Gegensatz zur physischen, wandelbar. Die Adressierung ist eng mit dem Routing (Wegfindung vom Sender zum Empfänger) verbunden. Dafür werden netzwerkübergreifende Adressen in Routingtabellen bereitgestellt und aktualisiert. Mithilfe dieser Tabellen werden Datenpakete an verfügbare Netzwerkknoten weitergeleitet, sodass die Datenpakete ihr Ziel finden.

Das Internetprotokoll (IP) ist in dieser Ebene angesiedelt.

3.1.4 Transportschicht / Transport Layer

Die Transportschicht ist das Bindeglied zwischen den transportorientierten und anwendungsorientierten Schichten. Die Datenpakete werden routingunabhängig abstrahiert und in standardisierten Ports den höheren Ebenen angeboten (z.B. ist Port 80 für das Protokoll HTTP im WWW reserviert). Nachfolgend werden die Datenpakete einer Anwendung zugeordnet. Wichtige Funktionen sind, dass Datenverluste erkannt und automatisch behoben werden und die Datenübertragung in beide Richtungen möglich ist.

In dieser Ebene operiert das Protokoll TCP als Transportmedium für das WWW, für E-Mail-Dienste und viele andere populäre Netzdienste.

3.1.5 Kommunikationsschicht / Session Layer

Die Kommunikationsschicht organisiert die Verbindungen zwischen den Endsystemen in sogenannten Sessions. Diese bestehen aus Anfragen und Antworten. In dieser Schicht werden Sicherungspunkte (Check Points) eingeführt, die es bei Verbindungsabbruch erlauben, die Sitzung nach Unterbrechung wieder aufzunehmen, ohne den gesamten Prozess der Datenübertragung von vorne starten zu müssen.

3.1.6 Darstellungsschicht / Presentation Layer

Die Darstellungsschicht wandelt die Daten in verschiedene Codecs (z.B. ASCII) und Formate um. Hier werden die Daten zu oder von der Anwendungsschicht in ein geeignetes Format umgewandelt. Diese Schicht übersetzt also zwischen systemabhängiger und systemunabhängiger Darstellung der Daten. Dazu gehören auch die Datenkompression und -verschlüsselung bzw. deren Umkehrung.

3.1.7 Anwendungsschicht / Application Layer

Die Anwendungsschicht stellt Funktionen für die Anwendungen zur Verfügung und ist in diese eingebettet. Auf dieser Ebene findet die Dateneingabe und -ausgabe statt. Die Anwendung selbst können darüber hinaus noch weitere Funktionen bereitstellen, die nicht der Datenübertragung und damit nicht dieser Schicht zugeordnet werden. Anwendungen sind beispielsweise Webbrowser, E-Mail-Programme, Instant Messaging Dienste und Smartphone Apps. In dieser Schicht ist z.B. das Protokoll https angesiedelt. Es stellt Websites dar und kann mithilfe von Links zu beliebigen Servern navigieren und deren Daten als Websites darstellen.

3.2 Übertragungsstandards in lokalen Netzwerken

Nachfolgend werden Protokolle beschrieben, die häufig von vernetzten Softwareprodukten genutzt werden und in den untersuchten Beispielen vorkommen. Diese Protokolle können mehrheitlich den transportorientierten Schichten zugeordnet werden und erstrecken sich über mehrere Schichten.

3.2.1 Bluetooth

Bluetooth beinhaltet mehrere Protokolle, die in zwei Stapel (Stacks) sortiert werden können (Controller und Host Stack) und sich über mehrere OSI Schichten bis zur Kommunikationsschicht erstrecken. Bluetooth kommt oft bei der kabellosen Einbindung von peripheren Geräten (z.B. Lautsprecher, Computermaus etc.) zum Einsatz. Ein wichtiger Entwicklungsschritte war 2010 der Standard 4.0. Dazu gehört der Modus Low Energy, der es ermöglicht, kleine Geräte mit geringer Batterie-Kapazität mit Bluetooth Chips auszustatten. 2016 wurde der Standard 5.0 veröffentlicht, der eine höhere Datenrate und Reichweite (bis zu 100m) ermöglicht. Version 5.1 und 5.2 benötigen beide neue Hardware, alle Versionen sind aber abwärtskompatibel, sodass neue Standards von alten „verstanden“ werden.

3.2.2 Thread

Wesentliche Charakteristika des Protokolls sind eine geringe Leistungsaufnahme und ein Parallelbetrieb mit anderen Sendern auf den 2,4 und 5 GHz Frequenzen, insbesondere WLAN und Bluetooth. Thread nutzt das Protokoll UDP in der Transportschicht, 6LoWPAN (eine Version des IP) in der Vermittlungsschicht und den Standard IEEE 802.15.4 (wireless protocol mit mesh communication) in den Schichten 1 und 2. Thread beinhaltet kein Application Layer. An dieser Stelle kann z.B. ZigBee aufsetzen.

3.2.3 ZigBee

ZigBee 3.0 ist ein Standard, der Home Automation und LL (Light Link) zusammenführt. Es baut ebenfalls auf dem Standard IEEE 802.15.4 in Schicht 1 und 2 auf. Mehrere Protokolle in den darüberliegenden fünf Schichten erweitern dessen Funktionalität insbesondere um die Möglichkeit des Routings und des sicheren Schlüsselaustausches. Beispielsweise in der Netzwerkschicht nutzt ZigBee unter anderem die Protokolle 6LoWPAN und RPL (routing protocol for low power and lossy networks). Das 6LoWPAN-Protokoll ermöglicht die direkte Anbindung von Geräten oder Sensornetzwerken an das Internet (über IPv6).

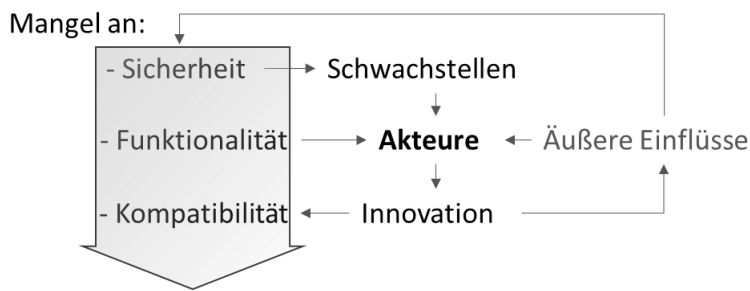
3.2.4 Matter

Es handelt sich bei Matter um ein IP-Protokoll zur Datenübertragung in der Vermittlungsschicht, das Wi-Fi (802.11) und Thread (802.15.4) unterstützt und zur Inbetriebnahme Bluetooth Low Energy benötigt. Im Jahr 2021 werden die ersten Produkte mit dem neuen Standard erwartet (Trust 2021). Der neue Standard soll vor allem eine unterbrechungsfreie lokale Kommunikation gewährleisten. Zudem soll er ermöglichen, kompatible Geräte in mehreren Smart-Home-Systemen ansprechen zu können, also Siri, Alexa oder Google Assistant.

3.3 Erweitertes Schichtenmodell und Abhängigkeitskategorien

Um das Schichtenmodell für die Analyse von softwarebedingter Obsoleszenz nutzen zu können, bedarf es einer Erweiterung um den Kontext von Akteuren und Einflussmöglichkeiten. Die Schichten des OSI Schichtenmodells bleiben dabei bestehen. Ihre Schnittstellen können jedoch im erweiterten Modell entlang der Wirkungskategorien Sicherheit, Funktionalität und Kompatibilität als Obsoleszenzfaktoren verstanden werden. Eingebettete Systeme sind modular aufgebaut und über Schnittstellen von mehreren Schichten abhängig. Gleichzeitig unterliegt jede Schicht selbst dem Einflussbereich verschiedener Akteure. Akteure können insbesondere Entwickler, Unternehmen, Betreiber von App Stores und ähnlichen Services oder auch Normungsgremien und Konsortien sein. Einzelne Schichten können sich unabhängig von anderen verändern. Dies geschieht meist aufgrund technischer Entwicklung, Innovationen oder veränderten Anforderungen und kann zur Obsoleszenz innerhalb der Schicht führen, z.B. durch neue Standards. Andererseits gefährdet jede Veränderung einer Schicht auch den Bestand der Schnittstellen zu anderen Schichten entlang der Wirkungskategorien, die in diesem Moment ebenfalls zu Obsoleszenzrisiken werden und das vorzeitige Nutzungs- oder Lebensende des eingebetteten Systems provozieren können. Die Wirkungskategorien sind zusammenfassend in Abbildung 8 dargestellt. Äußere Einflüsse sind vielfältiger Art und könnten z.B. veränderte Nutzungsgewohnheiten und -anforderungen an das Produkt sein. Die äußeren Einflüsse und ein Mangel an Sicherheit, Funktionalität oder Kompatibilität bewegen die Akteure zu Innovationen (z.B. Sicherheitsupdates), deren Software mindestens einer Schicht zuzuordnen ist. Diese können wiederum, auch mittelbar über äußere Einflüsse, Obsoleszenz entlang (einer) der Wirkungskategorien herbeiführen, sowohl innerhalb der Schicht als auch an einer Schnittstelle. Insgesamt führen veränderte äußere Faktoren (grüne Schrift in Abbildung 8) über ein kompliziertes Wirkungsgefüge mit Feedbackschleifen zu Obsoleszenz an den Schnittstellen zwischen den Schichten (orangene Schrift).

Abbildung 8: Wirkungsgefüge innerhalb einer Schicht



Obsoleszenz:

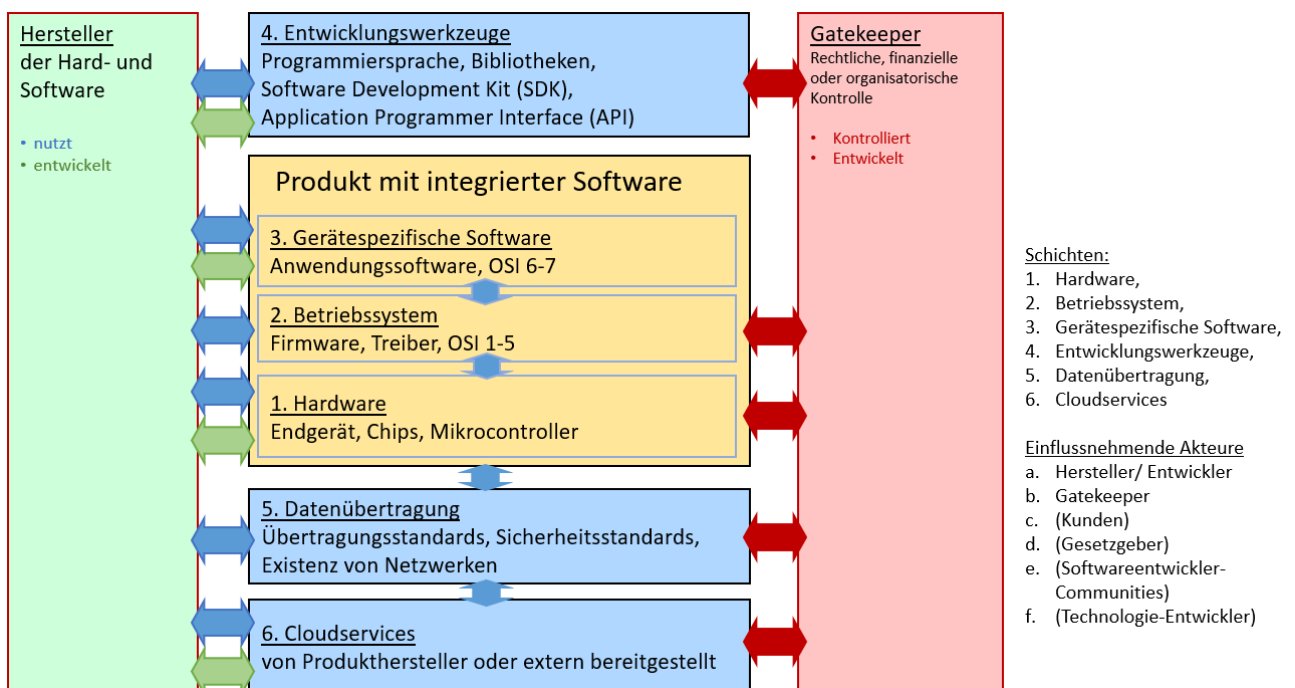
- Innerhalb der Schicht
- Schnittstelle zwischen den Schichten

Quelle: Eigene Darstellung

Die Analyse mithilfe dieses Schichtenmodells soll äußere Einflüsse finden, die Obsoleszenz verursachen.

Abbildung 9 gibt einen Überblick über die Einbettung und Wechselwirkung zwischen verschiedenen Schichten und Umgebungen. Gerätespezifische Software, Datenübertragung und Serverdienst (Cloudservice) stehen jeweils in unterschiedlichen Umgebungen und beinhalten jeweils mehrere Schichten, wobei jede Schicht wieder Akteure, Schnittstellen und Obsoleszenzrisiken enthält.

Abbildung 9: Schichtenmodell erweitert um Abhängigkeitsverhältnisse



Quelle: Eigene Darstellung, Öko-Institut e.V.

Um die Übersicht über dieses komplexe System zu behalten und das Modell sinnvoll auf Beispielsysteme anwenden zu können, nehmen wir folgende vereinfachende Aggregation vor. Dabei sollen insbesondere die funktionalen Abhängigkeiten der Software von Softwareentwickler oder Gerätehersteller im Fokus stehen. Wir unterscheiden in Abhängigkeiten der gerätespezifischen Software von:

1. der Hardware,
2. dem Betriebssystem,
3. der Datenübertragung,
4. der Entwicklungsumgebung,
5. Cloudservices,
6. Gatekeepern.

Die letzten beiden nehmen eine Sonderposition ein, da sie je nach Kontext mit dem Hersteller identisch sein können oder eine externe Abhängigkeit darstellen.

In der Mitte des Modells steht das physische Produkt (in der Abbildung 9 gelb hinterlegt), das sich aus Hard- und Software zusammensetzt. Die Software enthält alle für die Datenübertragung nötigen OSI Schichten, ggf. auch eine Anwendungsschicht. Integrierte Systeme sind meist über eine Datenübertragung mit einem Cloudservice und/oder einem anderen Gerät mit integrierter Software verbunden. Erst zusammen erbringt das gesamte Produktsystem die von ihm erwartete Funktionalität, insbesondere bei sogenannten smarten Anwendungen. „Smart“ ist jedoch in diesem Kontext kein wissenschaftlich definierter Begriff, sondern eine Selbstbezeichnung aus der Kundenkommunikation und deutet auf ein Geschäftsmodell hin, das einen Cloudservice als zentrale Steuereinheit beinhalten könnte. Die senkrechten Doppelpfeile weisen auf die starke funktionale Abhängigkeit zwischen den Schichten hin. Ein einziger Bruch kann zur Unbrauchbarkeit des gesamten Systems führen.

Gleichzeitig visualisiert das erweiterte Modell die Abhängigkeiten und Einflussnahme der Entwickler und die Einflussnahme der Gatekeeper. Je nach Kontext variiert die Rollenverteilung und der Entwickler hat mehr oder weniger Einfluss auf die einzelnen Schichten, die sie entweder nur nutzt oder auch selbst entwickelt und kontrolliert. Die drei untersuchten Beispiele sollen diese Variation abbilden und jeweils aus einem der Bereiche hoher Hard-/ Softwareabhängigkeit und starker Verschränkung von Hard- und Software stammen. Als Gatekeeper werden all solche externen Akteure bezeichnet, die rechtliche, finanzielle oder organisatorische Kontrolle über eine der Schichten haben. Diese Akteure sind oft Monopolisten bzw. Oligolisten oder haben eine sehr marktbeherrschende Stellung. Sie haben die Macht, anderen Marktakteuren die eigenen Bedingungen zu diktieren und den Zugang zu marktrelevanter Infrastruktur zu verwehren. Gatekeeper entstehen meist durch Plattform- und Netzwerkeffekte. Beispielsweise kann ein Gatekeeper bestimmte Entwicklungswerkzeuge bereitstellen, mit denen besonders einfach eine große Anzahl von Apps entwickelt werden können, die dann aber ausschließlich mit der Plattform desselben Anbieters funktionieren und somit die Popularität dieser Plattform fördern, wodurch diese finanziert wird. Theoretisch wäre es auch denkbar, dass die Datenübertragung von Gatekeepern kontrolliert wird. Über die sogenannte Netzneutralität wird durch eine Informationspflicht der Netzbetreiber über ihre Konfigurationen im Funkanlagen-gesetz (Bundesgesetzblatt 2017) Transparenz geschaffen.

3.4 Beispiel hoher Hardwareabhängigkeit: Smartphone App „Corona-Warn-App“

Die Corona-Warn-App bietet sich als Beispiel besonders an. Zum einen ist sie sehr bekannt. Sie ist (Stand Juli 2021) die am häufigsten heruntergeladene (kostenlose) App und war während ihrer Entwicklung Anfang 2020 Gegenstand breiter gesellschaftlicher Debatten. Mitte Juli 2021 wurde die App 31 Mio. mal heruntergeladen, davon 17,3 Mio. mal über den Google Play Store und 13,7 Mio. mal über den Apple App Store (Robert Koch-Institut 2021). Eine Schätzung des RKI geht davon aus, dass 75-80% der installierten Apps auch benutzt werden. Bisher wurden

479.824 (61% der teilbaren) Testergebnisse über die App geteilt (ebd.). Zum anderen ist sie transparent dokumentiert, was eine detaillierte Analyse ermöglicht.

Herausgeberin der App ist das Robert Koch-Institut (RKI), entwickelt wurde sie von den Unternehmen SAP und Deutsche Telekom AG unter Beteiligung von rund 25 weiteren Unternehmen. Der Quellcode ist öffentlich einsehbar (GitHub 2021) und das Projekt lädt Entwickler*innen dazu ein, sich bei der Programmierung einzubringen.

3.4.1 Schichten und Schnittstellen der Corona-Warn-App

Im Folgenden werden die Schichten einzeln auf die drei Obsoleszenzkategorien Sicherheit, Funktionalität und Kompatibilität hin untersucht.

3.4.1.1 Hardware

Die ältesten Geräte, bei denen die App installiert werden kann, sind das iPhone 5s (2013-2016) und Geräte, auf denen Android 6 (Marshmallow) lauffähig ist, das 2015 eingeführt wurde. Die App beansprucht etwa 13 MB/63MB Speicherplatz zur Installation der App unter Android/Apple iOS (Apple 2021b), (Google 2021). Für einige Funktionen ist der Zugriff auf die Kamera erforderlich (ebd.)

In Deutschland besitzen bei Einführung der App etwas mehr als 50 Millionen Menschen ein Smartphone, das technisch geeignet ist, die Corona-Warn-App zu nutzen. Der überwiegende Teil der Menschen, deren Smartphone für die Installation der App ungeeignet ist, ist älter als 65 Jahre (Scheuer 2020). Mehr als zehn Millionen Menschen im Alter über 65 besitzen gar kein Smartphone (Telecom 2020).

3.4.1.2 Betriebssystem/ ENS

Das Exposure Notification System (ENS) wurde von Apple und Google gemeinsam für die Kontaktnachverfolgung entwickelt und veröffentlicht (Apple und Google 2020). Das ENS ist im Betriebssystem von iOS und Android integriert. Es stellt die Programmierschnittstelle zwischen dem Betriebssystem und der Corona-Warn-App bereit. Apple hat das ENS über ein Update zur Verfügung gestellt, Google hat ENS ohne explizite Zustimmung der Nutzenden auf allen kompatiblen Geräten installiert. Das ENS unterstützt folgende Betriebssysteme:

- ▶ Android-Version 6.0 („Marshmallow“) oder neuer,
- ▶ iOS 12.5.1 oder neuer (ab 10. Februar 2021- zuvor war iOS 13.5 erforderlich).

Auf einem mobilen Gerät sendet das ENS einen regelmäßig wechselnden Code (sogenannte Rolling Proximity IDs) und sucht gleichzeitig mithilfe von Bluetooth in Abständen von 4 bis 5 Minuten jeweils ungefähr 4 Sekunden lang, ob andere Smartphones ebenfalls solche Kennungen aussenden, und speichert die Signalstärke und den Zeitpunkt des Empfangs für eine Dauer von 14 Tagen lokal ab. Die Suche nach Signalen und dessen Intervall ist Teil der vom Betriebssystem bereitgestellten Schnittstelle und kann nicht durch die App selbst verändert werden. Die RPIDs (Rolling Proximity IDs) werden bis zu viermal pro Sekunde verschickt. Die IDs sind nur für 10 bis 20 Minuten gültig und werden kryptografisch von temporären Schlüsseln abgeleitet, die sich alle 24 Stunden ändern. Nachdem ein mobiles Gerät die Liste aller vorhandenen Schlüssel der Personen vom Server heruntergeladen hat, deren positive Testung gemeldet wurde, leitet das Exposure Notification System (ENS) die jeweiligen IDs ab und prüft lokal, ob diese den lokal gesammelten Rolling Proximity Identifiern entsprechen. Damit die Software zuverlässig arbeiten kann, wird empfohlen, die Hintergrundaktualisierung und Bluetooth immer zu aktivieren und

dafür die Akkuverbrauchsoptimierung auszuschalten. Dies führt zu einem erhöhten Energieverbrauch und kürzeren Akkulaufzeiten.

Das Exposure Notification System (ENS) stellt potenziell ein erhebliches Risiko für den Schutz der Privatsphäre dar. Es ist anzunehmen, dass Apple und Google „durch eine Änderung des ENS zur Verknüpfung der dort verarbeiteten Tagesschlüssel und RPIs mit einer geräte- (z. B. Werbe-ID) oder nutzerspezifischen Kennung (z. B. Apple-ID oder Google-Konto) auf technischer Ebene leicht in der Lage wären“ (Robert Koch-Institut 2020). Diese Verknüpfung der persönlichen Daten (Apple-ID und Google-Konto) mit den pseudonymisierten Tagesschlüsseln würde den Mechanismus zum Schutz der Privatsphäre unterwandern und stellt somit ein Sicherheitsrisiko dar, das von Apple und Google ausgeht.

3.4.1.3 Gerätespezifische Software

Die Nutzenden erwarten hauptsächlich, dass die Corona-Warn-App erkennt, ob ein Kontakt mit einer mit Covid-19 infizierten Person stattgefunden hat. Die App nutzt für die Kontaktverfolgung einen dezentralen Ansatz, der auf dem Exposure Notification System aufbaut (GitHub 2021). Wird eine Nutzer*in positiv auf das Coronavirus getestet, kann er oder sie das Ergebnis eines kooperierenden Testlabors über einen QR Code in der App registrieren. Die Schlüssel der letzten 14 Tage der positiven Nutzenden werden daraufhin an den Server übermittelt und der Liste angefügt, die regelmäßig an alle Apps verschickt wird. Im Kontaktfall wird das Risiko des Kontakts durch die App anhand der lokal gespeicherten Metadaten und vom RKI festgelegter Parameter eingeschätzt und die Person erhält entsprechende Handlungsanweisungen.

Weitere Funktionen der App sind die Anzeige von Statistiken (z.B. 7-Tage-Inzidenz oder die Anzahl der bisher vollständig geimpften Menschen) und das Erstellen eines QR-Codes für Events und Geschäfte. Außerdem existieren eine Funktion zum Check-in, die auf die Kamera zugreift und ein Kontakt-Tagebuch, sowie eine Funktion zum Verwalten der EU Covid-Zertifikate, die Auskunft über den Impf-, Test- und Genesenenstatus ausgeben können.

Nach dem Zufallsprinzip werden unechte Meldungen erstellt, gesendet und dann im Backend des Cloudservices wieder verworfen. Diese unechten Meldungen sind von den gültigen Meldungen nicht zu unterscheiden und erzeugen eine Art Hintergrundgeräusch, durch das der Sendeaufruf der TANs und das Hochladen der Infektionsdaten nicht von diesem Hintergrundgeräusch unterscheidbar sind. So kann selbst bei überwachtem Netzwerkverkehr eine plausible Abstreitbarkeit geschaffen werden.

3.4.1.4 Datenübertragung

Das ENS nutzt den Funkstandard von Bluetooth Low Energy, um Signale mit anderen Geräten auszutauschen. Die Bluetooth Schnittstelle basiert auf dem Privacy-Preserving Contact Tracing Protokoll (PPCP). Dabei ist die Suche nach Signalen und dessen Intervall Teil der vom Betriebssystem bereitgestellten Schnittstelle. Solange in der App die Risiko-Ermittlung aktiviert und Bluetooth eingeschaltet ist, werden Begegnungen vom Exposure Notification System (ENS) von iOS/Android aufgezeichnet. Die App selbst muss dafür nicht geöffnet bleiben.

Die „grobe Standortermittlung“ anhand von WLAN- oder Mobilfunkdaten muss bei einigen Geräten (speziell allen Android-Versionen vor Version 11) aktiviert sein, da Bluetooth-Geräte in der Nähe bei diesen Geräten nur gefunden werden können, wenn die „grobe Standortermittlung“ auf dem Gerät aktiviert ist. Dies kann zu einem erhöhtem Energieverbrauch führen.

Insbesondere bei Android nutzen dadurch auch andere Apps die Funktion der Standortermittlung, was den Energieverbrauch und das Sicherheitsrisiko erhöht.

3.4.1.5 Entwicklungsumgebung

Apps für iPhones müssen von Apple verifiziert und über den Store vertrieben werden. Der Prozess der Überprüfung der Anwendung dauert etwa 3-4 Wochen. Apple legt dabei Kriterien an Design-Principles und Funktionalität. Apple steht es frei, Apps willkürlich abzulehnen (Giga 2017). Das Apple Developer Program kostet 99 USD pro Mitgliedschaft und Jahr (Apple 2021a). Das notwendige Software Development Kit (SDK) namens XCode existiert ausschließlich für MacOS, das Apple Betriebssystem für Computer. Laut den Lizenzbedingungen muss der Code in Objective-C, C++ oder C geschrieben werden. Das User Interface muss in Swift programmiert werden.

Android-Apps laufen in Java (Yeeply 2020). Sie werden in dem SDK namens Android Studio geschrieben.

3.4.1.6 Cloudservice

Die Deutsche Telekom betreibt für die Corona-Warn-App vier Server:

- ▶ den Corona-Warn-App-Server. Er speichert die autorisierten Meldungen positiver Corona-Tests, so lange sie nicht älter als 14 Tage sind.
- ▶ den Verifikations-Server. Er dokumentiert, dass ein Nutzender freiwillig zugestimmt hat, die Meldung über seinen positiven Corona-Test einzureichen, und dass diese Meldung vom Labor als zutreffend bestätigt („verifiziert“) wurde.
- ▶ den Portal-Server. Er erzeugt teleTANs für noch nicht digital an das App-System angeschlossene Gesundheitsämter und Labore.
- ▶ den Testergebnis-Server. Er stellt die Ergebnisse der Labore zur weiteren Verwendung bereit.

Die Verteilung der Daten auf mehrere Server soll dazu dienen, diese Daten zu entkoppeln und so eine Rückverfolgung zu einzelnen Nutzenden zu erschweren. Seitdem Mitte Oktober 2020 ein europäischer Gateway-Service in Betrieb gegangen ist, ist auch eine länderübergreifende Risiko-Ermittlung möglich. Derzeit funktioniert mit der Corona-Warn-App ab Version 1.15.1 die Risiko-Ermittlung mit 17 Ländern⁸.

Ende-zu-Ende-Verschlüsselungen wie HTTPS stellen sicher, dass Meldungen während der Datenübertragung nicht gelesen werden können. Der Corona-Warn-App Server empfängt die Schlüssel aller positiven Testergebnisse, die jünger als 14 Tage sind, und verteilt diese regelmäßig an alle Apps. Das ENS berechnet daraus kryptographisch die RPIDs zurück und vergleicht diese mit den lokal gesammelten.

Nutzende müssen für die richtige Funktion der App auch diesen Datenschutzbestimmungen zustimmen. Positive Testergebnisse können unter Umständen nicht in die App eingetragen werden. Dieser Fall liegt vor, wenn der Nutzende an der Teststation der Erlaubnis zur Datenübermittlung nicht zugestimmt hat (Klaus 2020). In diesem Beispiel wird der Zielkonflikt zwischen Datenschutz und Funktionalität deutlich.

⁸ Belgien, Dänemark, Finnland, Irland, Italien, Kroatien, Lettland, Litauen, Niederlande, Norwegen, Österreich, Polen, Schweiz, Slowenien, Spanien, Tschechien und Zypern

3.4.1.7 Gatekeeper

Das ENS erlaubt ausschließlich staatlichen Gesundheitsbehörden oder Regierungen Corona-Apps für die Betriebssysteme iOS und Android zu entwickeln. Hierfür ist eine Serverinfrastruktur erforderlich, die die positiven Testergebnisse verifiziert. Die Betriebssysteme für Smartphones von Apple und Google sind so aufgebaut, dass die Installation von Apps standardmäßig über einen App Store funktioniert. Auf diese Weise kontrollieren Apple und Google den Markt an angebotener Software und setzen Qualitäts- und Sicherheitsstandards durch. Dies hat zur Folge, dass Nutzende sich bei diesen Stores ebenfalls registrieren müssen. Bei Android sind dies die proprietären Google Play Dienste (oder bei neueren Huawei-Geräten aus Lizenzgründen die Huawei Mobile Services) oder die Open-Source-Implementierung der Google-Play-Dienste microG. Bei Apple ist ein Apple-Benutzerkonto zur Nutzung des App Stores erforderlich. Die jeweiligen Datenschutzbestimmungen sind zu akzeptieren.

3.4.2 Politik der Softwareunternehmen zur Unterstützung älterer Hardware

Normalerweise ist ein Softwareprodukt, das für ein bestimmtes Betriebssystem geschrieben wird, auf eine Version dieses Betriebssystems zugeschnitten und unterstützt ältere Betriebssysteme nicht. Dies resultiert aus einer einfachen Kosten-Nutzen-Rechnung, bei der sich der Support mit Updates und die Entwicklungskosten nicht mehr lohnen, wenn die Marktdurchdringung der älteren Betriebssystemversion einen gewissen Schwellenwert unterschreitet. Bei dem Beispiel der Corona-Warn-App ist das Softwareunternehmen jedoch keinem großen Preisdruck ausgesetzt. Dennoch konnte die App bei iPhones zunächst nur auf dem neuesten Betriebssystem installiert werden. Dies liegt daran, dass die App auf dem Exposure Notification Framework aufgebaut ist, das im Betriebssystem von iOS verankert ist und von Apple erst ein Jahr später für ältere Geräte zur Verfügung gestellt wurde. Apple unterstützt Geräte, die älter als 5 Jahre sind, im Allgemeinen nicht mehr mit neuer Software.

Bei Android konnte Google den Google Play Service bei allen Smartphones, die den Bluetooth-Standard Bluetooth Low Energy unterstützen, aktualisieren und somit auch bei dem nicht mehr unterstützten Betriebssystem Android 6 (Marshmallow) aus dem Jahr 2015 das Exposure Notification Framework einführen.

3.4.3 Zusammenhang zwischen der Datensicherheit und der jeweiligen Software-Version über den Zeitverlauf

Insbesondere in agiler Softwareentwicklung ist es gängige Geschäftspraxis, in festgelegten Zeitintervallen funktionsfähige Zwischenstände des Softwareproduktes zu veröffentlichen und an die Bedürfnisse der Nutzenden und Auftraggeber anzupassen. Einerseits wird der Funktionsumfang erweitert, wodurch die Anforderungen an die Hardware zunehmen, und andererseits können Sicherheitslücken geschlossen werden. Das Verhältnis von Fehlerbehebungen (bug fixes), Sicherheits- und funktionellen Updates über den Zeitverlauf gibt einen Eindruck der Softwarequalität in Hinblick auf Erweiterbarkeit und Sicherheit. In diesem Beispiel dominieren Funktionserweiterungen. Ausschließlich bei Update Version 1.5 der Corona-Warn-App handelt es sich um ein Sicherheitsupdate (CWA Open Source Projekt 2021). Die geringe Anzahl an Sicherheitsupdates kann dadurch erklärt werden, dass durch den dezentralen Ansatz Security by Design gegeben ist und die Offenheit des Codes Fehler offenbart, bevor Updates freigegeben werden.

3.4.4 Übersicht

Von den oben beschriebenen Schichten der Corona-Warn-App lassen sich entsprechend des Wirkungsgefüges Risiken der softwareinduzierten Obsoleszenz ableiten.

Gerätehersteller vergrößern normalerweise ihren Absatz, wenn die Dauer der Kompatibilität mit Anwendungssoftware verkürzt wird. Dadurch wird funktionsfähige Hardware obsolet. In diesem Beispiel jedoch wurden bestehende Betriebssysteme mit der ENS Schnittstelle nachgerüstet. Der limitierende Faktor in der Abwärtskompatibilität ist bei Android u.a. der Bluetooth Chip für 4.0, dessen Vorgängerversionen für die Hauptfunktion der Corona-Warn-App technisch ungeeignet sind. Warum Apple die Unterstützung auf Geräte ab Baujahr 2013 beschränkt, ist aus technischer Sicht nicht ersichtlich und kann in dieser Studie nicht beantwortet werden.

Das Betriebssystem induziert bei Gebrauch des ENS eine erhöhte Batteriebeanspruchung, da die Ortungsfunktion aktiviert und von anderen Diensten zusätzlich genutzt wird. Dies führt zu kürzeren Ladezyklen und damit zu einer schnelleren Abnutzung des Akkus. Durch den zunehmenden Funktionsumfang steigen die Ansprüche an die Hardware. Eine (empfundene) Verlangsamung der Software kann zu psychologischer Obsoleszenz führen kann. Die Datenübertragung ist durch ein Sicherheitsrisiko gefährdet, indem die RPIs leicht mit persönlichen Daten verknüpft werden könnten, sollten sich die Datenschutzbestimmungen des ENS ändern. Die offene Entwicklung der App beugt vielen Risiken vor, da bei einer großen Entwicklercommunity Gemeinnützigkeit gegenüber Einzelinteressen der Vorzug gegeben wird. Allerdings ist das Gesamtsystem sehr von Gatekeepern abhängig. Diese üben insbesondere über das im Betriebssystem integrierte ENS einen großen Einfluss auf die App aus.

Nachfolgende Tabelle 6 fasst die Obsoleszenzrisiken des Gesamtsystems jeweils bezogen auf die beteiligte Systemschicht zusammen.

Tabelle 6: Obsoleszenzrisiken Beispiel 1: Smartphone Corona-Warn-App

Schicht	Ursachen für Obsoleszenzrisiken	Höhe des Obsoleszenzrisikos
Hardware	Ein Teil des Betriebssystems (ENS) ist nur für bestimmte (neue) Hardware verfügbar und zwingende Voraussetzung für die App	Sehr hoch
Betriebssystem	Induziert erhöhte Batteriebeanspruchung	Hoch
Gerätespezifische Software (App)	Performanz und psychologische Obsoleszenz, Sicherheitsupdates (in geringem Maß) notwendig	Gering
Datenübertragung	Sicherheit (Datenschutz könnte sich ändern)	Mittel
Entwicklungsumgebung	Gerätespezifische Software ist offen	Sehr gering
Cloudservice	Solide finanzierte Kooperation	Gering
Gatekeeper	kontrolliert weite Teile des Gesamtsystems: Hardware, Betriebssystem, Details der App (über den App Store, Entwicklungsumgebung und Betriebssystem), Bluetooth, Datenübertragung, Entwicklungsumgebung	Sehr hoch

3.5 Beispiel starker Verschränkung von Hard- und Software: „Smarte Leuchtmittel mit Steuerungszentrale“ Philips Hue

Smarte Leuchtmittel haben standardisierte Schraubfassungen und können dadurch in bestehende Lampen oder Lampenfassungen eingebaut werden. Neben einem LED-Leuchtmittel enthalten die „smarten“ Leuchtmittel eine Datenschnittstelle, über die sie mit Hilfe einer Fernbedienung oder eines Smartphones über das Heimnetzwerk gesteuert werden können.

Neben den Funktionen An und Aus sind diese meist dimmbar und können ihre Farbtemperatur und/oder die Farbe wechseln.

Hier betrachten wir das Produktsystem des Herstellers Philips, der für seine Produkte in diesem Bereich auch die beiden Markennamen Signify und Hue verwendet. Das Gesamtsystem besteht aus einem Sender (Hue Bridge), der über LAN mit einem Router verbunden ist, den Leuchtmitteln, einer Smartphone App, einem Cloudservice und ggf. einem Sprachassistenzsystem, smarten Lichtschaltern und weiteren smart Home Geräten.

3.5.1 Schnittstellen und Schichten von Philips Hue

3.5.1.1 Hardware

Die Steuerungsoberfläche aller „smarten“ Geräte ist eine App für Smartphones. Seit 2019 werden smarte Leuchtmittel mit einem Chip für Bluetooth 5 verkauft, das 2016 eingeführt wurde. Das Leuchtmittel ist auf eine Nennlebensdauer von 25.000 Stunden ausgelegt. Die Leuchtmittel sind zusammen mit dem Sender (Hue Bridge) als Starterset zu erwerben.

3.5.1.2 Betriebssystem

Das Betriebssystem für Leuchtmittel und Sender wird von Phillips entwickelt. Updates werden über das Internet bereitgestellt und müssen über die Bridge auf dem Leuchtmittel installiert werden. Das REST (Representational State Transfer) Application Programming Interface (API) ist offen zugänglich (<https://developers.meethue.com/develop/hue-api/>) und kann genutzt werden, um die Hue Bridge ohne die App direkt (z.B. über einen Browser) anzusteuern. REST API nutzt http Anfragen, um der Endsoftware zu sagen, was sie tun soll (Philips Hue 2021a).

Die App erfordert iOS 13+, das auf iPhone 6s oder neuer (Markteinführung 2015) läuft. Für Android ist Version 8.0 (2017) oder neuer erforderlich. Die API kann auch genutzt werden, um beispielsweise einen RaspberryPi als Steuerung zu programmieren.

3.5.1.3 Datenübertragung

Die Leuchtmittel kommunizieren mit der Hue Bridge per ZigBee (siehe Abschnitt 3.2). Seit 2012 wird der ZigBee Light Link Standard verwendet. Ab 2018 wurde ZigBee auf Version 3.0 umgestellt, die durch ein Update auf der Bridge und den Leuchtmittel installiert werden konnte und abwärtskompatibel ist.

Befindet sich das Smartphone im selben lokalen Netzwerk wie die Bridge, wird die LAN-Verbindung zwischen Router und Bridge genutzt. Mit dem Smartphone können die Leuchtmittel auch direkt über Bluetooth kommunizieren. Der Funktionsumfang mit Bluetooth ist kleiner als über ZigBee, insbesondere die Anzahl der Leuchtmittel ist begrenzt, die gesteuert werden kann (10 statt 50) (Philips Hue 2022).

Die Bridge benötigt über den Router eine Internetverbindung, wenn sie auch außerhalb des Heimnetzes angesteuert werden soll oder wenn Updates heruntergeladen werden sollen. Die Hauptfunktionalitäten funktionieren auch ohne Internet.

3.5.1.4 Gerätespezifische Software

Über die App wird die API der Bridge angesteuert. Dadurch werden die Ressourcen (lights, groups, config, schedules, scenes, sensors, rules) verwaltet und entsprechende Parameter können gelesen und gesetzt werden.

3.5.1.5 Entwicklungsumgebung

Den Autor*innen ist nicht bekannt, welches Software Development Kits (SDK) und welche Programmiersprache von Hue für die Programmierung des Betriebssystems der Leuchtmittel und der Bridge genutzt wird.

3.5.1.6 Cloudservice

Für die Steuerung der Leuchtmittel von unterwegs ist der Cloudservice von Hue über das Internet zwischengeschaltet. Nachdem die Bridge auf dem Cloud-Server „Hue Server“ registriert wurde, kann man mit einer Userkennung über das Internet auf die API der Bridge zugreifen ⁹.

3.5.1.7 Verknüpfung mit weiteren smart Home Systemen

Die API der Hue Bridge kann auch von verschiedenen Smart Home-Plattformen und -Geräten genutzt (etwa Apple HomeKit, Google Home, Amazon Alexa) werden, sodass das System zusätzlich durch Sprache gesteuert werden kann.

Soll die Hue Bridge in die Sprachassistenten Alexa, Google Assistant, Samsung Smart Things oder Apple HomeKit eingebunden werden, ist eine Registrierung bei Hue und Verknüpfung der beiden Cloudservices notwendig. Beispielsweise muss bei Alexa die Installation des entsprechenden „Skills“ (Sprachbefehls) im Alexa-System erfolgen. In der Alexa App müssen anschließend die Hue Kontodaten eingegeben werden. Um die Hue Bridge in das HomeKit System einzubinden, benötigt man eine Bridge v2, ein iOS Gerät mit Version 9.3+, die HomeKit App, die Hue App und eine Internetverbindung. Der Aufdruck der Bridge enthält einen Registrierungscode für HomeKit, über den die Bridge in HomeKit eingebunden wird. Auf diese Weise erfolgt die Steuerung vom Sprachassistenten über die Server desselben zum Clouddienst von Hue, welcher der Bridge die Anweisungen weitergibt.

Bestimmte Geräte, wie z.B. Amazons Echo Plus, beinhalten eine ZigBee Schnittstelle, die die Grundfunktionen der Hue Bridge übernehmen können. Allerdings können Softwareupdates nur über die Hue Bridge auf den Lampen installiert werden. Die zwischengeschalteten Geräte, von Google Home oder die verschiedenen Ausführungen des Amazon Echos, setzen zwingend eine Verbindung zur Hersteller-Cloud voraus. Dort erst werden nämlich die Befehle und Anfragen der Nutzenden analysiert, dann in Aktionen übersetzt und an den Hue Clouddienst weitergeleitet.

3.5.1.8 Gatekeeper

Für seine eigenen Smarten Leuchtmittel kontrolliert Philips sowohl die Hardware als auch das Betriebssystem der Leuchtmittel und der Bridge. Die API ermöglicht anderen Herstellern, das System um Funktionalitäten und Geräte zu erweitern. Die offiziellen Smartphone Apps werden über die beiden marktbeherrschenden Plattformen zur Verfügung gestellt und unterliegen den dortigen Abhängigkeiten (siehe Abschnitt 3.4.1.7).

3.5.2 Supportzeiträume, Update-Politik und technischer Fortschritt

Signify (Philips) hat bereits angekündigt, alle Hue-Leuchtmittel mit dem Funkstandard (auch Verbindungsstandard genannt) Matter kompatibel machen zu wollen. Der Funkstandard Matter ermöglicht die kompatible Steuerung von Smarthome-Produkten. Philips will ein entsprechendes Update über die Hue-Bridges an die angeschlossenen Geräte verteilen (Költzsch 2021). Lampen ohne Bridge können kein Update erhalten und werden weiterhin ausschließlich über Bluetooth vom Smartphone angesteuert. Die Bluetooth-Funktion der Lampen wurde 2019 eingeführt.

⁹ https://api.meethue.com/bridge/<whitelist_identifier>

Es werden für mindestens ein Jahr Softwareupdates für wichtige Funktionen der Bridge bereitgestellt. Für die Bridge gewährleistet Hue die Kompatibilität mit ihren Online-Diensten und der neuesten Version der mobilen Hue App für einen Zeitraum von mindestens drei Jahren, für die Leuchtmittel für einen Zeitraum von fünf Jahre.

Der Support der Hue Bridge 1 (erstmalig 2012) wurde 2020 nach sieben Jahren eingestellt, die aktuelle Version ist v2.1 (Version 2.0 kam 2015 auf den Markt). Die Version 1 ist technisch veraltet, da sie nicht in der Lage ist, Apple HomeKit (Markteinführung 2014) anzusteuern (Wernicke 2019). Das Supportende hat zur Folge, dass die Möglichkeit wegfällt, von unterwegs über Philips Clouddienste die Steuerung vorzunehmen. Die Lampen behalten weiterhin im lokalen Netzwerk ihre gesamte Funktion und Signify veröffentlichte eine spezielle App für die Bridge v1 (ebd.).

Philips Hue begründet das Supportende auf seiner Website mit dem Argument der Qualität der Nutzererfahrung und rät gleichzeitig zum Neukauf:

„Um zu gewährleisten, dass unsere Kunden weiterhin die bestmögliche Beleuchtungsleistung erhalten, hat Signify beschlossen, weitere Software-Updates und Online-Dienste für die Hue Bridge Version 1 einzustellen [...] Signify kann nach dem Serviceende die Sicherheit der Hue Bridge Version 1 nicht mehr gewährleisten. [...] Sollten Softwareupdates bzw. die Kompatibilität mit Online-Diensten und/oder mobilen Apps eingestellt werden, empfehlen wir Dir, eine neue Version des Produkts zu erwerben.“ (Philips Hue 2021b).

3.5.3 Übersicht

In diesem Beispiel werden zusätzlich zu dem eigentlichen Produktsystem ein Router und ein Smartphone mit entsprechendem Betriebssystem vorausgesetzt, sodass diese Geräte dem Risiko von Obsoleszenz ausgesetzt sind, sollten sich die Anforderungen ändern. Sicherheitsupdates der Leuchtmittel und der neue Funkstandard Matter können ausschließlich über die Bridge installiert werden, sodass ggf. Sicherheitsrisiken und Kompatibilitätsprobleme entstehen, wenn diese nicht zu Verfügung steht. Ebenso kann in diesem Fall der Clouddienst nicht genutzt werden, sodass die Funktionalität eingeschränkt ist. Die lokale Grundfunktionalität ist jedoch weiterhin bei allen Generationen der Leuchtmittel gegeben, die zudem untereinander keine Kompatibilitätsprobleme aufweisen (David 2018). Das Betriebssystem der Bridge stellt eine offenen API bereit, die eine große Erweiterbarkeit ermöglichen. Dadurch können existierende und sparsame Smart Home Geräte genutzt und eingebunden werden, andererseits besteht für Konsument*innen der Anreiz, mehr davon nachzufragen (z.B. ein externes Smart Home System). Welcher Effekt im Sinne des Ressourcenverbrauches schwerer wiegt, kann im Rahmen dieser Studie nicht beantwortet werden. Das Produktsystem ist von externen marktbeherrschenden Plattformen und deren Standards abhängig. Deren Änderung (Markt-ein/aus-tritt) kann zur Obsoleszenz des Produktsystems führen, wie das Beispiel HomeKit zeigt. Nachfolgende Tabelle 7 fasst die Obsoleszenzrisiken der einzelnen Schichten zusammen.

Tabelle 7: Obsoleszenzrisiken Beispiel 2: Smarte Leuchtmittel

Schicht	Ursachen für Obsoleszenzrisiken	Höhe des Risikos
Hardware	Router und Smartphone benötigt	Mittel
Betriebssystem	Support von Sicherheitsupdates	Mittel
Gerätespezifische Software (App)	Kompatibilität zum Smartphone	Mittel

Schicht	Ursachen für Obsoleszenzrisiken	Höhe des Risikos
Datenübertragung	Sicherheit, neuer Standard (Matter) wird über Softwareupdate eingeführt.	Gering
Entwicklungsumgebung	Offene API	Sehr gering
Cloudservice	Verfügbarkeit, Sicherheitsupdates werden über die Cloud der Bridge zur Verfügung gestellt.	Hoch
Externes Smart Home System	Wenige marktbeherrschende Anbieter, Service nicht garantiert	Hoch
Gatekeeper	Die Einführung von Apple HomeKit hat zu Veralterung der Bridge v1 geführt	Mittel

3.6 Beispiel hoher Softwareabhängigkeit: Smartes Heizkörperthermostat

Die Hauptfunktionalität eines smarten Heizkörperthermostaten ist es, die Raumheizung zu regulieren. Dabei sollen zusätzliche Informationen, die über das Internet zugänglich sind (z.B. das Wetter), mit lokalen Informationen (z.B. der Raumtemperatur) verknüpft werden. Die User-Erwartung, die durch Werbung und Herstellerversprechen generiert wird, ist eine Effizienzsteigerung und eine Reduktion der Heizkosten im Vergleich zu herkömmlichen Heizkörperthermostaten.

Es sollen hier solche Geräte untersucht werden, die sich in marktrelevante Plattformen integrieren lassen. Der smarte Thermostat ersetzt Heizkörperthermostate mit Anschlussgewinde M30 x 1,5mm (es gibt Adapter für andere Ventiltypen). Für die Schnittstelle zwischen dem eigentlichen Gerät und dem Router werden SmartHome Steuerungseinheiten benötigt. Die zwei Beispiele sind so ausgewählt, dass sie ausschließlich mit einer Steuerungseinheit a) eines anderen Herstellers (Eve Thermo) und b) desselben Herstellers (tado° Smartes Heizkörper-Thermostat) funktionieren.

3.6.1 Schichten und Schnittstellen von Eve Heizkörperthermostaten

3.6.1.1 Hardware

Neben dem Thermostaten werden ein mobiles Endgerät von Apple und für die Außerhaus-Steuerung ein Router und ein Smart Home Gerät von Apple benötigt. Dazu zählen HomePod, iPad oder AppleTV (4. Generation oder neuer). Das untersuchte Gerät ist die dritte Generation seines Typs. Ihr wurden im Vergleich zur zweiten Version zwei neue Funktionalitäten hinzugefügt.

3.6.1.2 Betriebssystem

Der Thermostat ist kompatibel mit HomeKit für iPhone mit iOS 12.4 oder iPad mit iPadOS 13 (beide September 2019) oder neuer. Die Eve App erfordert allerdings iOS 14.2 (November 2020) oder neuer für mobile Geräte und macOS 11.0 (oder neuer) mit einem Mac mit Apple M1-Chip für Computer. Firmwareupdates werden über die Eve App installiert.

3.6.1.3 Datenübertragung

Die Datenschnittstelle ist jeweils über den aktuellen Bluetooth Standard von HomeKit gewährleistet. HomeKit benutzt ein Kommunikationsprotokoll von Apple (HomeKit Accessory

Protocol (HAP)), das mit Wifi und Bluetooth Protokollen arbeitet. Im Thermostat ist ein Chip für Bluetooth 4.0 verbaut, neue Geräte von Apple sind mit Bluetooth 5.0 und Thread (siehe Abschnitt 3.2) ausgestattet. Bluetooth verschiedener Versionen sind miteinander kompatibel.

3.6.1.4 Gerätespezifische Software

Die Eve App und Apple HomeKit stellen die Funktionalitäten des smarten Thermostaten bereit. Dazu gehört das Setzen der Zieltemperatur und das Erstellen von Zeitplänen. Der Thermostat erkennt einen schnellen Abfall der Temperatur als offenes Fenster und die Abwesenheit des steuernden Smartphones und sperrt entsprechend das Heizungsventil.

3.6.1.5 Entwicklungsumgebung

Über die internen Prozesse der Programmierung des Betriebssystems ist den Autor*innen nichts bekannt. Die API wird nicht öffentlich dokumentiert oder unterstützt.

3.6.1.6 Cloudservice

Ist der Thermostat mit dem Cloudservice von Apple (iCloud) gekoppelt und ein Gerät als Heimzentrale eingerichtet, übernimmt dieses die Steuerung und ist auch außer Haus über den Cloudservice von Apple (iCloud) ansteuerbar. Der Fernzugriff über die Heimzentrale ist per Ende-zu-Ende-Verschlüsselung gesichert, und die Daten werden ausschließlich auf dem iPhone (oder iPad) gespeichert.

3.6.1.7 Gatekeeper

Das Gerät ist von Apple als „works with HomeKit“ zertifiziert. Dafür muss es das MFI (ursprünglich „made for iPod“) Programm durchlaufen und unterliegt Kriterien und Anforderungen, die Apple festlegt. Dazu gehören z.B. Anforderungen an die Datenverschlüsselung und die Authentifikation.

3.6.2 Schichten und Schnittstellen von Tado Heizkörperthermostaten

3.6.2.1 Hardware

Neben den Heizkörperthermostaten wird die Internetbridge desselben Herstellers benötigt. Diese ist mit einem LAN-Kabel an einen Router angeschlossen, über den die Bridge mit einem Cloudservice kommuniziert. Das Heizkörperthermostat ist über eine Smartphone App bedienbar. Die Apple Version der App erfordert iOS 13.0 oder neuer und 73 MB Speicherplatz.

3.6.2.2 Betriebssystem

Verschiedene Chipsätze benötigen unterschiedliche Firmwareversionen. Diese werden automatisch installiert, sobald das Gerät Verbindung zum Cloudservice aufgebaut hat.

3.6.2.3 Datenübertragung

Die Bridge kommuniziert über 868 MHz (6LoWPAN, ist gleich IPv6) mit dem Thermostaten. Die Bridge fungiert ausschließlich als Funkmodul zur Internetanbindung aller Tado Geräte in der Wohnung.

3.6.2.4 Gerätespezifische Software

Da die Daten im Cloudservice verarbeitet und gespeichert werden, dient die App lediglich als Visualisierung der Steuerungseinheit und benötigt für die Steuerung einen Internetzugang. Der volle Funktionsumfang wird gegen eine monatliche Gebühr freigeschaltet. Dazu gehört das Erkennen von offenen Fenstern und der Anwesenheit von Personen.

3.6.2.5 Entwicklungsumgebung

Über die internen Prozesse der Programmierung des Betriebssystems ist den Autor*innen nichts bekannt. Die API wird nicht öffentlich dokumentiert oder unterstützt.

Es gibt allerdings Open Source Projekte, die eine API nachbauen (Openhab 2021).

3.6.2.6 Cloudservice

Die App ermöglicht die Steuerung über den Cloudservice mit den drei marktbeherrschenden Home Assistenz Diensten Google Assistant, Apple HomeKit und Alexa.

3.6.3 Standardisierungsgrad und Offenlegung der Schnittstellen

Eve hat angekündigt, mit der vierten Generation den Funkstandard Thread zu unterstützen, das Apple Ende 2020 in seinem HomePodmini eingeführt hat (Evehome 2021). Die vorherigen Generationen von 2016, 2017 und 2019 werden den neuen Funkstandard nicht unterstützen. Thread ist jedoch abwärtskompatibel. Zukünftig werden Apple und Eve den Funkstandard auf Matter umstellen. Der Eve Thermostat wird weiterhin ausschließlich mit dem Apple Produkt-Ökosystem kompatibel und geschlossen sein.

Auch Tado lässt keine Absicht erkennen, die API zu veröffentlichen.

3.6.4 Übersicht

Eve: Dieses System ist hauptsächlich geprägt durch die Abhängigkeit von einer Plattform. Apple kontrolliert die Hardware, das Betriebssystem, den Standard der lokalen Datenübertragung, die Datenübertragung von Außerhaus, die Entwicklungsumgebung und die Modalitäten der Bereitstellung der App. Dies führt dazu, dass ausschließlich die neuesten iPhones unterstützt werden. Hat sich ein Nutzender für dieses Produktsystem entschieden, wird das Obsoleszenzrisiko für das Smartphone erhöht, wenn er nicht im Besitz der aktuellen Version des iPhone ist. Technisch notwendig scheint die ausschließliche Unterstützung der neuesten iPhone Generation nicht zu sein, eine erhebliche negative Wirkung auf die Ressourceneffizienz hat sie jedoch. Andererseits hat Apple die Möglichkeit, hohe Standards an Qualität und Sicherheit durchzusetzen.

Tabelle 8: Obsoleszenzrisiken Beispiel 3a: Smartes Thermostat Eve

Schicht	Ursachen für Obsoleszenzrisiken	Höhe des Risikos
Hardware	iPhone der neusten Generation und Router notwendig	Sehr hoch
Betriebssystem (Homekit)	Nicht abwärtskompatibel	Hoch
Gerätespezifische Software (App)	Nicht kompatibel mit vorrangegangenen Betriebssystemen	Sehr hoch
Datenübertragung	Neuer Standard (Matter) wird über Softwareupdate eingeführt (aktuelle Eve App notwendig).	Gering
Entwicklungsumgebung	Geschlossene API,	Mittel
Cloudservice	iCloud für Steuerung von unterwegs	Gering
Gatekeeper	Apple legt Bedingungen fest für: Funkstandards, Kompatibilität, Funktionalität, Sicherheit, Anforderungen an die App	Hoch

Tado: Die Obsoleszenzrisiken ergeben sich aus internen organisatorischen Abhängigkeiten, insbesondere der Verfügbarkeit des Cloudservice, der in diesem Beispiel der Hauptbestandteil des Produktsystems ist. Da die API geschlossen ist, kann der Service bei Ausfall auch nicht durch eine andere Organisation übernommen werden.

Tabelle 9: Obsoleszenzrisiken Beispiel 3b: Smartes Thermostat Tado

Schicht	Ursachen für Obsoleszenzrisiken	Höhe des Risikos
Hardware	Router und Smartphone benötigt	Gering
Betriebssystem	Chipsätze benötigen unterschiedliche Firmwareversionen,	Gering
Gerätespezifische Software (App)	Kompatibel mit iOS 14 und Android	Gering
Datenübertragung	Sicherheit (Datenschutz), Verfügbarkeit des Internetanschlusses,	Gering
Entwicklungsumgebung	Geschlossene API	Mittel
Cloudservice	Verfügbarkeit, Sicherheitsupdates werden über die Cloud der Bridge zur Verfügung gestellt.	Hoch
Externes Smart Home System	Wenige marktbeherrschende Anbieter, Service nicht garantiert	Gering
Gatekeeper	-	Sehr gering

Dieses Produktsystem zeichnet sich durch relativ geringe externe Abhängigkeit aus. Hauptsächlich bestehen diese im AppStore und bei der Verfügbarkeit des Cloudservice. Der Hersteller wäre theoretisch in der Lage, eine lebenslange Garantie für die Funktionalität des Gesamtsystems zu geben. (Da er für die volle Funktionalität eine monatliche Gebühr nimmt, ist er in diesem Fall nach der Warenkaufsrichtlinie ab 1.1.2022 dazu verpflichtet, für die Zeit der Nutzung die Vertragsmäßigkeit des Produktes sicherzustellen.)

3.7 Zusammenfassung der Wirkungszusammenhänge von Soft- und Hardware

Immer mehr Produkte, die Software enthalten, sind auf vielfältige Weise in ein System eingebettet, durch das sie ihre volle Funktionalität erhalten. Oftmals wird die Bedienoberfläche auf das Smartphone verlagert und über eine Funkverbindung oder über einen Cloudservice über das Internet zur Verfügung gestellt. Das Produkt eines Herstellers erhält seine Funktionalität oft sogar erst durch den Service eines anderen Unternehmens. Dies erfordert eine große Konnektivität, die zu einer Vielzahl von Schnittstellen führt, die von unterschiedlichen Akteuren kontrolliert werden. Mangelnde Interoperabilität und **Kompatibilität**, insbesondere zwischen verschiedenen Generationen und Standards, bilden ein besonders hohes Risiko für softwarebedingte Obsoleszenz von Hardware. Die Abhängigkeiten, die verkürzend auf die Lebensdauer einwirken, lassen sich in sechs Kategorien unterteilen. Diese sind Abhängigkeiten von:

1. der Hardware z.B. physische Schnittstellen, Funkchips, Endgeräte

2. der Datenübertragung (OSI Schicht 2-5), z.B. Funkstandards, Verschlüsselungstechnologien
3. dem Betriebssystem (OSI Schicht 6-7),
4. der Entwicklungsumgebung
5. Cloudservices und von der Rechenzentrums Umgebung, sowie
6. Gatekeepern und Plattformen.

Zusätzlich können interne organisatorische Abhängigkeiten, beispielsweise Updatehäufigkeiten, Supportzeiträume und die Laufzeit eigener Cloud-Services, zur Obsoleszenz der am Produktsystem beteiligten Geräte führen.

Deshalb sollten Produkte, die Software enthalten, als Produktsystem verstanden werden, die zusammen eine **Funktionalität** bereitstellen. Für diese komplexen Systeme lassen sich oft Kernfunktionalitäten formulieren, die ein Nutzender von diesem Produktsystem erwartet. Diese sind je nach Anwendungsfall unterschiedlich. In unseren drei Beispielen könnten dies folgende Kernfunktionalitäten sein:

- ▶ Corona-Warn-App: Registrierung von Begegnungen, Austausch der Schlüssel mit einem Server, Benachrichtigung im Falle einer Begegnung mit einer infizierten Person
- ▶ Philips Hue: Bedienen und Fernsteuern von Leuchtmitteln
- ▶ Eve und Tado Heizkörperthermostat: Fernsteuern der Thermostate und Erstellen von Zeitplänen für den Heizkörper.

Diese Kernfunktionalitäten könnten von den Herstellern des Produktes definiert werden. Bei Geräten mit viel Gestaltungsspielraum in der Anwendungsschicht (z.B. Smartphone) wandelt sich die erwartete Funktionalität sehr schnell. Es hat sich etabliert, dass Updates neue Funktionalitäten integrieren und somit zur Hardwaredegradation und zu psychologischer Obsoleszenz beitragen.

Ein weiteres hohes Obsoleszenzrisiko stellt die Wahrung der **Sicherheit** dar. Mit Zunahme der Schnittstellen steigt auch die Wahrscheinlichkeit einer böartigen Infiltrierung an diesen Schnittstellen. Hier ergibt sich ein Zielkonflikt mit der Kompatibilität, der sich technisch beispielsweise durch kabelgebundene Verbindungen lösen lässt. Der Markt und das Design der Produkte entwickeln sich jedoch bei Smart Home Produkten in die Richtung, dass alle Geräte denselben Funkstandard bedienen, nämlich Matter. Dieser wird für Hersteller attraktiver werden, je mehr andere Dienste auf diesen Funkstandard setzen.

Gerätehersteller haben ein ökonomisches Interesse daran, dass keine Sicherheitslücken in ihren Produkten auftauchen. Dies kann durch Security by Design und qualitativ hochwertigen Code eingeschränkt, jedoch niemals gänzlich ausgeschlossen werden. Es ist jedoch gängige Geschäftspraxis, noch nicht gänzlich ausgereifte Software auszuliefern und durch Updates nachzubessern.

4 Untersuchung der Marktentwicklung

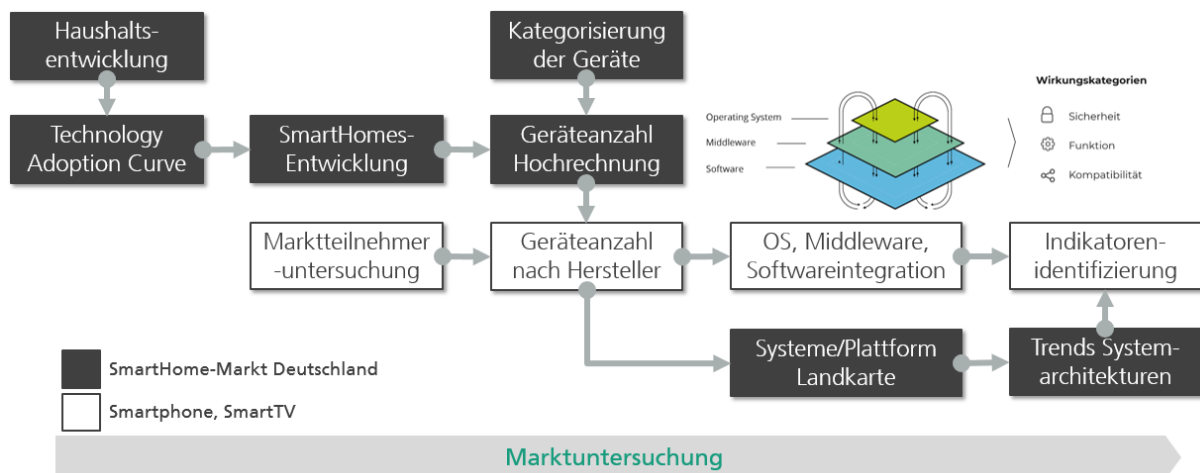
Im Fokus dieses Arbeitspakets steht die Untersuchung der Marktentwicklung von vernetzten Haushaltsprodukten bis zum Jahr 2030. Eine große Herausforderung besteht hier in der fehlenden allgemein gültigen Definition des Begriffs „SmartHome“. Das hat zur Folge, dass unterschiedliche Quellen diesen Produktbereich und die darin eingeschlossenen Geräte häufig auch unterschiedlich interpretieren. Damit sind ihre Ergebnisse nur begrenzt transparent und schwer vergleichbar. Aus diesem Grund wird hier auch nur eine orientierende Hochrechnung erstellt, die in den folgenden Detailuntersuchungen des Abschnitts 4.4 vertieft werden. Wichtig bei der Marktbetrachtung ist der Projektfokus auf die drei Wirkungskategorien Funktion, Sicherheit und Kompatibilität, die wesentlich für die Bewertung von Obsoleszenz-relevanten Risiken smarter Produkte sind.

Ergebnis dieser Teiluntersuchung ist eine detaillierte Darstellung der Marktentwicklung von Produkten im Bereich SmartHome auf Basis verschiedener Quellen und Marktforschungsunternehmen und -instituten. Dieses Vorgehen liefert eine systematische Übersicht aller großen Marktteilnehmer, deren Produkte und Marktvolumina. Anhand ausgewählter Indikatoren erfolgt eine Risikobewertung, ob und wie Software auf die Nutzungsdauer der Produkte reduzierend einwirkt.

4.1 Fragestellungen und Vorgehensweise

Nach den ersten Marktrecherchen mit dem Fokus auf die softwaretechnische Basis im SmartHome-Bereich (bspw. Betriebssystem, Programmiersprachen, Frameworks, Schnittstellen und Softwareintegration) hat sich gezeigt, dass dieses Vorgehen aufgrund der Vielfalt der Geräte und der vielen Überschneidungen mit den genannten Spezifika nicht praktikabel ist. Beispielsweise werden Betriebssysteme wie Android TV herstellerübergreifend genutzt (z.B. Phillips, Sony, Hitachi, Toshiba). Des Weiteren werden herstellerbezogene Betriebssysteme wie WebOS (LG) oder Tizen (Samsung) bei Geräten mehrerer Kategorien (SmartTV, SmartWatch, Smartphone) eingesetzt. Dieser Umstand wirkt sich allerdings auch positiv auf die weiteren Rechenschritte aus, da er den zu untersuchenden Gerätepool nicht nur sinnvoll begrenzt, sondern auch Erkenntnisse über Gerätekategorien liefert, die aus dem Untersuchungsrahmen des Projektes ausgeschlossen sind. Aus diesem Grund wurde ein Vorgehensmodell (Abbildung 10) entwickelt, das den Fokus der Marktanalyse auf die relevanten Marktteilnehmer (Amazon, NEST, Honeywell) richtet. So kann der Einfluss auf die softwarebedingte Obsoleszenz viel genauer spezifiziert werden. Wird zum Beispiel eine bestimmte Schnittstelle nicht mehr weiterentwickelt, kann die Anzahl der betroffenen Produkte und Produktsysteme besser abgeschätzt werden, die dann als nicht mehr sicher gelten.

Abbildung 10: Vorgehensmodell für die Marktanalyse im Bereich SmartHome



Quelle: Eigene Darstellung

Wie in Abbildung 10 dargestellt, wird die Untersuchung der Haushaltsentwicklung an den Beginn der Marktuntersuchung gestellt und mit den Daten der Technologieakzeptanzkurve (Bohlen 1957) korreliert, um die Entwicklung der SmartHome-Haushalte bis zum Jahr 2030 abzubilden. In Kombination mit der Gerätekategorisierung erfolgt anschließend eine grobe Abschätzung der aktuellen und zukünftigen Geräteanzahl in deutschen Haushalten. Relevante Marktteilnehmer werden anhand ihrer Marktanteile identifiziert. Diese Hochrechnungen sind Basis für zwei weitere Elemente der Marktuntersuchung. Einerseits soll eine System-Landkarte entstehen, die einen Überblick der Plattformen auch über die marktdominierenden Anspruchsgruppen hinaus liefern soll. So werden auch Trends in den Systemarchitekturen identifizierbar sein, die (noch) keine signifikante Marktrelevanz besitzen. Des Weiteren soll sie die Untersuchung der Betriebssysteme, Middleware und Softwareintegration und der daraus resultierenden Identifizierung der Indikatoren unterstützen. Diese Indikatoren liefern Rückschlüsse auf die genannten Wirkungskategorien Sicherheit (z.B. Updatezyklen verwendeter Frameworks), Funktion und Kompatibilität (z.B. unterstützte Geräte oder Schnittstellen). Auf diese Weise kann eine Abschätzung des Einflusses einer z.B. sicherheitsrelevanten Veränderung auf die SmartHome-Geräte in den deutschen Haushalten getroffen und spezifiziert werden. Eine Vertiefung dieser Arbeitsschritte erfolgt im Abschnitt 5.

Zusammenfassend konzentriert sich das Vorgehensmodell auf die folgenden Schwerpunkte:

- ▶ Untersuchung der Haushaltsentwicklung in Deutschland bis zum Jahr 2030
- ▶ Hochrechnungen für die Anzahl der Geräte pro Haushalt und die Anzahl der smarten Haushalte bis 2030
- ▶ Untersuchung von Marktvolumen (bisherige Wachstumsraten) und Marktpotential (erwartete Wachstumsraten)
- ▶ Kategorisierung vernetzter Haushaltsgeräte und Marktteilnehmer im Bereich SmartHome
- ▶ Kurze Betrachtung des SmartTV- und Smartphone-Segments in Bezug auf Marktanteile und Marktteilnehmer sowie technischer Spezifikationen von SmartTVs
- ▶ Ableitung von Indikatoren für eine grobe Einordnung des Obsoleszenzrisikos der untersuchten Produktkategorie

- Darstellung der Kompatibilitätsverhältnisse verschiedener Systeme/Plattformen

4.2 Haushaltsentwicklung in Deutschland

Im ersten Schritt der Marktuntersuchung muss das Potential von SmartHome-Anwendungen in Deutschland zunächst quantifiziert werden. Dazu wurde die Entwicklung der Privathaushalte in Deutschland bis 2030 untersucht (Abbildung 11).

Abbildung 11: Entwicklung der Haushalte in Deutschland 2019 und 2030

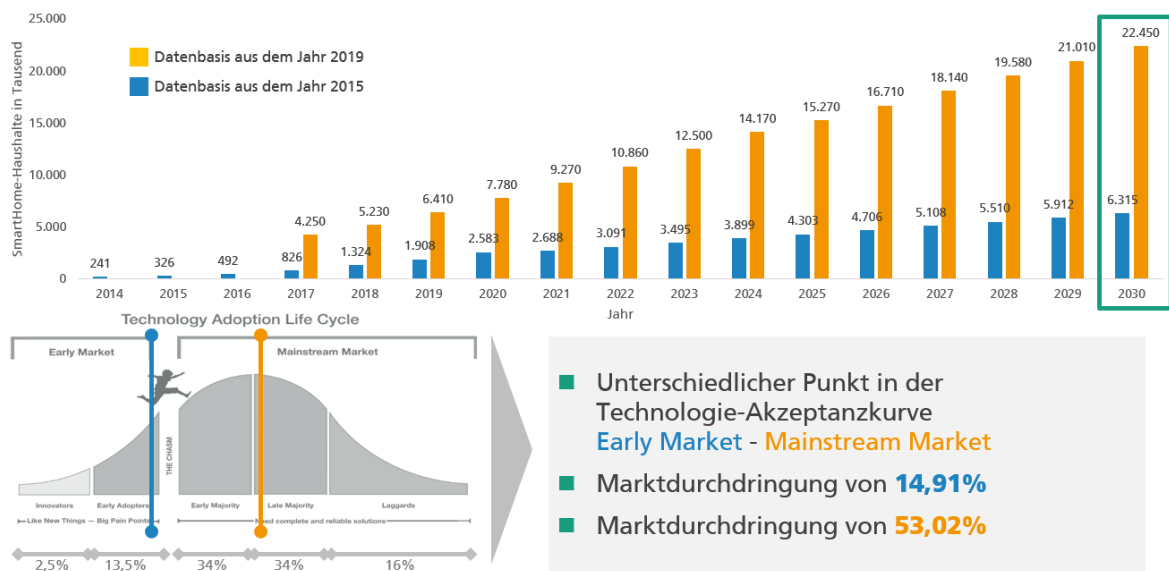
Haushalte für Deutschland 2019 bis 2030*

Haushaltsgröße	Haushalte 2019		Haushalte 2030		
Einpersonenhaushalte	17.557.000	42,3 %	18.357.000	43,4 %	▲
2 - Personenhaushalte	13.781.000	33,2 %	14.239.000	33,6 %	▲
3 - Personenhaushalte	4.952.000	11,9 %	4.676.000	11,0 %	▼
4 - Personenhaushalte	3.783.000	9,1 %	3.675.000	8,7 %	▼
Haushalte mit 5 Personen und mehr	1.434.000	3,5 %	1.396.000	3,3 %	▼
Insgesamt	41.506.000	100,0 %	42.343.000	100,0 %	▲

Quelle: Eigene Abbildung mit Daten von Destatis [Statistisches Bundesamt] (2020), Entwicklung der Privathaushalte bis 2040

Bis zum Jahr 2030 wird eine steigende Anzahl bei den Ein- und Zweipersonenhaushalten prognostiziert. Einpersonenhaushalte werden in Bezug auf das Jahr 2019 um 1,1% und Zweipersonenhaushalte um 0,4% zunehmen. Haushalte mit drei oder mehr Personen gehen hingegen zurück. Insgesamt wird ein Zuwachs von rund 837.000 Haushalten im Vergleich zu 2019 (41.506.000 Haushalte) bis zum Jahr 2030 (42.343.000 Haushalte) erwartet. Mit Hilfe dieser Haushaltsanzahl wird nun eine Hochrechnung der erwarteten smarten Haushalte erstellt. Die wenigen Datenquellen bzw. Marktentwicklungsstudien zu SmartHome-Geräten prognostizieren teilweise stark variierende Ergebnisse zum Wachstum und zur Marktdurchdringung (BMW 2016). In Abbildung 12 werden nun zwei Prognosedatensätze von Statista aus dem Jahr 2015 und 2019 verglichen und die fehlenden Datenpunkte bis zum Jahr 2030 linear extrapoliert.

Abbildung 12: Entwicklung der smarten Haushalte in Deutschland bis 2030



Quelle: Statista 2019, Baker Tilly 2015

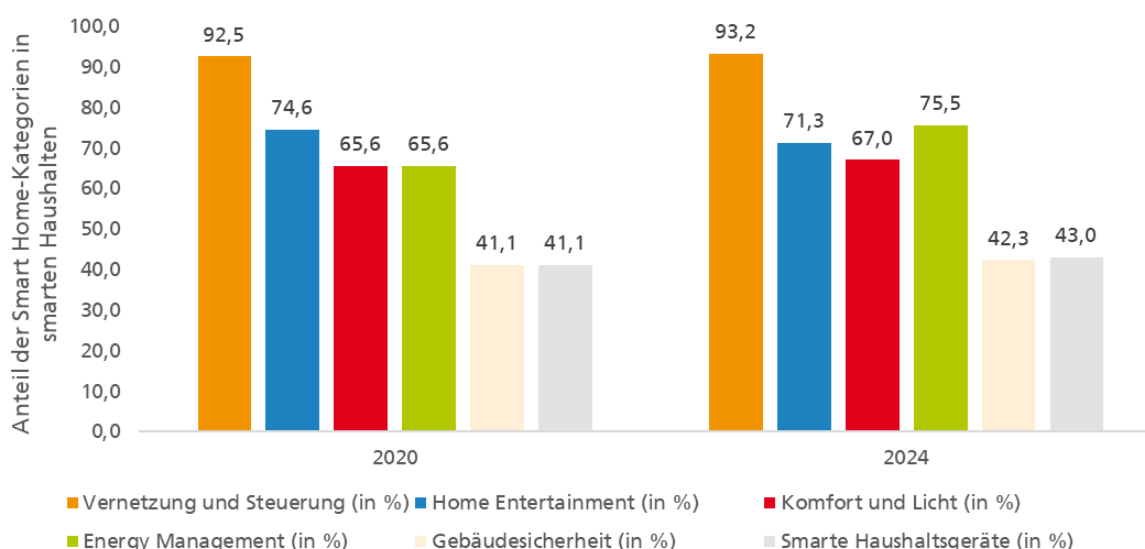
Nach der Auswertung der beiden Statista-Datensätze in Korrelation mit der herausgestellten Haushaltsentwicklung in Deutschland wurde eine große Varianz festgestellt. Unter der Annahme eines linearen Wachstums werden abhängig von der gewählten Quelle bis zum Jahr 2030 zwischen 6,3 Mio. und 22,5 Mio. Haushalte smart sein. Vergleicht man nun diese beiden Ergebnisse mit der Technologieakzeptanzkurve, ist von einer Marktdurchdringung von 14,91% (Early Market) bzw. 53,02% (Mainstream Market) Marktdurchdringung von SmartHome-Geräten im Jahr 2030 auszugehen. Hier ist die große Spannweite gut zu erkennen, die Prognosen in diesem volatilen Marktsegment aufweisen und in nur einen wenigen Jahren nach oben korrigiert wurden. Es ist allerdings eher davon auszugehen, dass SmartHome-Produkte im Jahr 2030 bereits im Mainstream Markt angekommen sein werden.

4.3 Hochrechnung der Geräteanzahl in smarten Haushalten

Die bereits festgestellten starken Unterschiede in den Prognosen selbst innerhalb eines Marktforschungsunternehmens im Bereich SmartHome erschweren genaue Hochrechnungen für die Geräteanzahl. In einer Studie SmartHome2Market aus dem Jahr 2016 vom BMWi wurden fünf Marktuntersuchungsinstitutionen und deren erwartete Wachstumsraten des globalen SmartHome-Marktes verglichen. Hier bezieht sich das Wachstum auf das Referenzjahr 2012 und prognostiziert bis 2020. Die Ergebnisse reichen von knapp 280% (MarketsandMarkets) bis hin zu über 1000% (Transparency Market Research). Im weiteren Vorgehen wird die Betrachtung des Anteils der smarten Haushalte (Statista 2020) in Kombination mit der in 4.2 erstellten Haushaltshochrechnung für die Hochrechnung der Geräteanzahl verwendet, auch da sich globale Prognosen nur bedingt auf Deutschland beziehen lassen.

Im Mai 2020 hat Statista für das Jahr 2020 und 2024 die Anzahl der Haushalte geschätzt, die über SmartHome-Anwendungen verfügen. Zusammen mit der Haushaltshochrechnung (Abschnitt 4.2) ergeben sich Erkenntnisse zum Anteil der SmartHome-Kategorien in smarten Haushalten (Abbildung 13) und der Anteil der SmartHome-Kategorien in allen Haushalten (Abbildung 14).

Abbildung 13: Anwendungen in smarten Haushalten in Deutschland



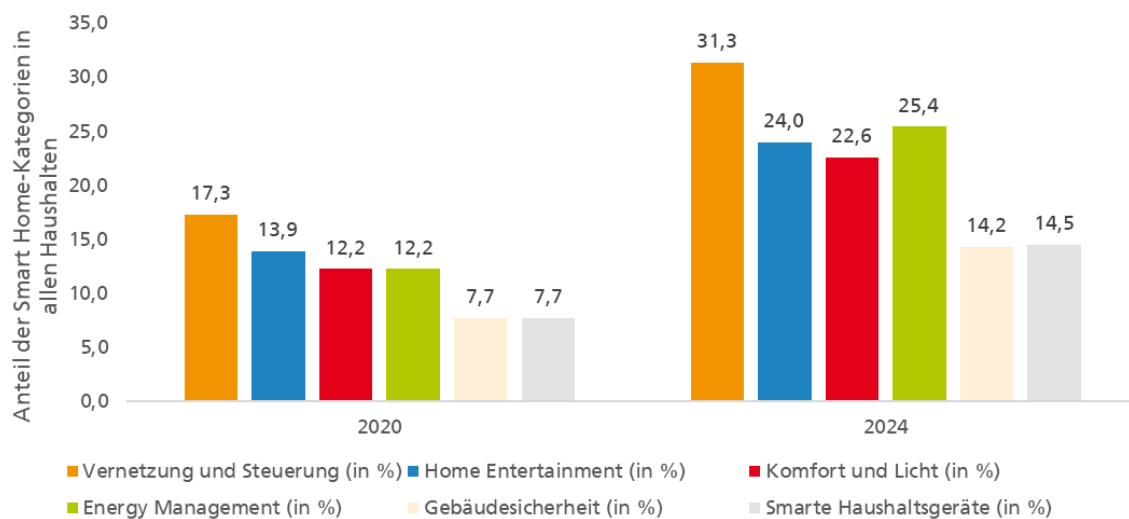
Quelle: Eigene Darstellung

Mit 92,5% sind Geräte zur Vernetzung und Steuerung (bspw. Smart Hubs, intelligente Steckdosen, Status Monitore) in smarten deutschen Haushalten in 2020 am stärksten vertreten. Dieser Anteil wird sich bis 2024 auf 93,2% erhöhen. Home Entertainment liegt bei 74,6% und verringert sich auf 71,3% bis 2024. Der Bereich Komfort und Licht erhöht sich von 65,6% auf 67%. Den größten Zuwachs gibt es laut Statista bei der Geräteklasse Energy Management. Ihr Anteil steigt um fast 10%. Smarte Gebäudesicherheits- und Haushaltsgeräteeinrichtungen sind mit jeweils 41,1% (2020) am geringsten vorhanden. Diese Werte erhöhen sich hingegen bis 2024 um 1,2% bzw. 1,9%. Das bedeutet, dass es im Jahr 2020 mind. 29,6 Mio. Geräte und im Jahr 2024 mind. 55,6 Mio. Geräte in Deutschlands smarten Haushalten vorhanden sind.

Aufgrund der hohen Anwendungsvielfalt und Varianz der unterschiedlichen Systemlösungen wird hier von einer Gleichverteilung der absoluten Gerätezahlen ausgegangen. Sind laut Statista 10,1 Mio. Geräte im Bereich Home Entertainment (2024) vorhanden, wird auch von 10,1 Mio. Haushalten mit einer solchen Anwendung ausgegangen. Bei 14,2 Mio. smarten Haushalten ergibt sich damit ein Anteil von 71,3% in der Kategorie Home Entertainment. Wie sich diese absoluten Gerätezahlen tatsächlich auf die smarten Haushalte verteilen ist nur schwer abzuschätzen. Gerade im Bereich Vernetzung und Steuerung oder Komfort und Licht ist davon auszugehen, dass eine Systemanwendung aus mehreren Elementen und mindestens einer zentralen Steuereinheit besteht. Zum Untersuchungszeitpunkt lagen dazu keine Informationen vor und machen diese Vereinfachung notwendig.

Unter Einbezug der steigenden Haushaltsanzahl in Deutschland zeigt Abbildung 14 wie sich der Anteil von SmartHome-Anwendungen in Bezug auf alle und nicht nur smarte Haushalte verteilt.

Abbildung 14: Anwendungen in allen Haushalten in Deutschland



Quelle: Eigene Darstellung

Sind im Jahr 2020 in 17,3% der deutschen Haushalte Geräte zur Vernetzung und Steuerung vorhanden, steigt dieser Anteil bis 2024 bereits auf 31,3%. Der Bereich Home Entertainment steigt um 10,1%, der von Komfort und Licht um 10,4%, Energie Management um 13,2%, Gebäudesicherheit um 6,5% und smarte Haushaltsgeräte um 6,8%. Den stärksten Zuwachs von 14% ist demnach bei Vernetzung und Steuerung zu beobachten.

Um diese Berechnungen zu validieren gibt es nur wenige bis keine relevanten Quellen. Die Bitdefender Studie (Bitdefener 2016) geht davon aus, dass im Jahr 2016 bereits zehn smarte Geräte pro Haushalt in Deutschland vorhanden waren. Nehmen wir nun die Werte von Statista aus dem Jahr 2020 und korrelieren diese mit den berechneten Haushaltsdaten des Statistischen Bundesamtes, müssten mind. 29,6 Mio. smarte Geräte in deutschen Haushalten vorhanden sein. Das sind im Jahr 2020 zwar elf Geräte pro smartem Haushalt aber nur 0,71 Geräte pro Haushalt insgesamt. Für das Jahr 2024 sind es 14 Geräte pro smartem Haushalt und 1,32 Geräte auf alle Haushalte gerechnet. Diese Abweichung um den Faktor 10, die vermutlich auf die uneinheitliche Definition des Begriffs „SmartHome“ zurückzuführen ist, hätte einen wesentlichen Fehler auf eine mögliche Hochrechnung für das Jahr 2030 zur Folge. Auf eine solche Hochrechnung wird an dieser Stelle verzichtet, da sie weder aussagekräftige noch belastbare Zahlen und Erkenntnisse liefert.

Zusammenfassend ist hier aber bereits das große Wachstumspotentials des SmartHome-Marktes in Deutschland sichtbar. Gerade aufgrund der stark steigenden Haushaltsanzahl und der stetig steigenden Marktdurchdringung dieser Produktkategorien.

4.4 Marktuntersuchungen ausgewählter Produkte

Nach der Abschätzung des Marktpotentials von Smarten Geräten in deutschen Haushalten folgt nun eine nähere Betrachtung des SmartTV- und des Smartphone-Marktes. Sie dienen als repräsentative Beispiele für die Kategorie Vernetzung und Steuerung sowie Home Entertainment, die neben dem Bereich des Energy Managements und Komfort und Licht, die als am stärksten wachenden SmartHome-Anwendungsfelder identifiziert wurden. Als ersten Schritt wird eine Kategorisierung der im SmartHome-Markt angesiedelten Geräte diskutiert. Am Ende des Abschnitts 4.4 erfolgt eine nähere Betrachtung der Systeme und Plattformen sowie der Anbieter in den verschiedenen Marktsegmenten. Dabei wird auch eine grobe Bewertung dieser

Anbieter vorgenommen, um das Potential möglicher Konflikte in Bezug auf die Wirkungskategorien Sicherheit, Kompatibilität und Funktion und damit softwarebasierten Einflussnahme auf eine verkürzte Nutzungsdauer vorgenommen.

4.4.1 Kategorisierung von SmartHome-Geräten

Wie bereits im Abschnitt 4.3 gezeigt, ist eine Definition und damit die Kategorisierung je nach Quelle sehr unterschiedlich, da einige Studien z.B. das Smartphone oder den Desktop-Computer als SmartHome-Geräte in ihre Betrachtung mit einbeziehen und andere nicht. Grundsätzlich wurden vier mögliche Kategorisierungen für SmartHome-Geräte identifiziert:

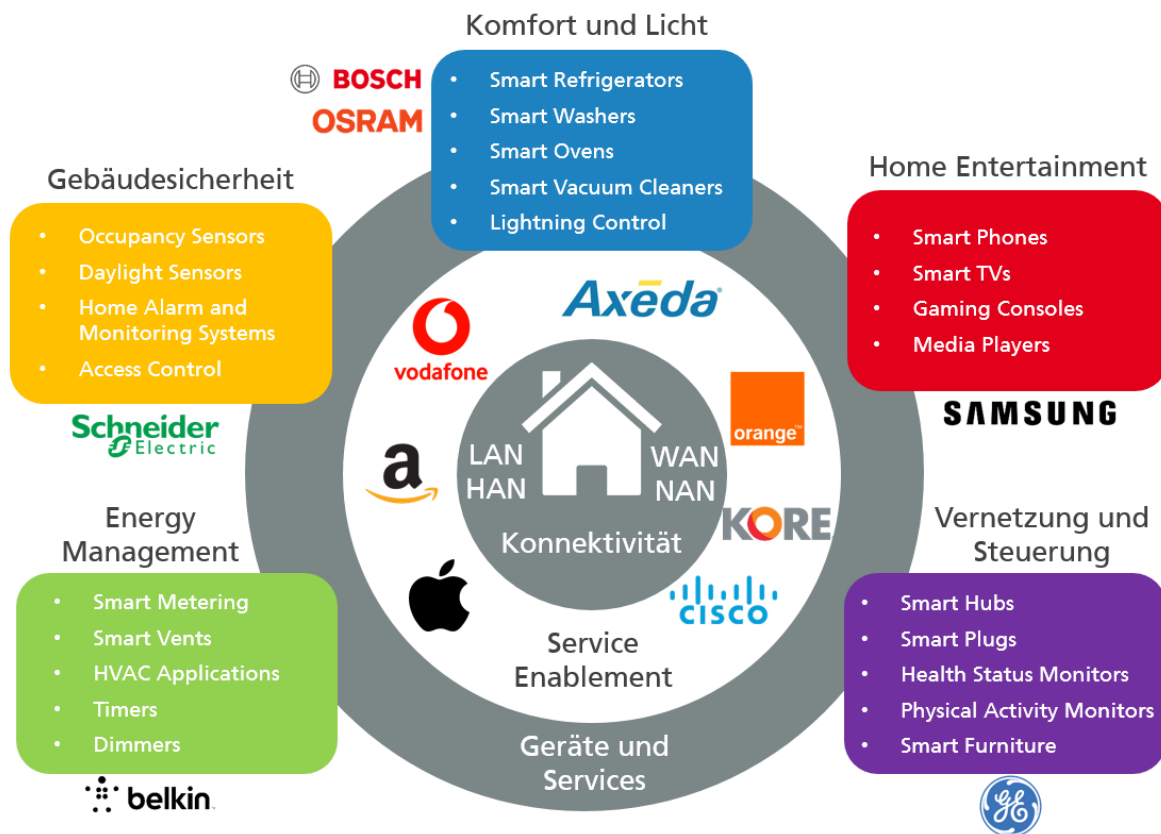
Tabelle 10: mögliche Kategorisierungen für SmartHome-Geräte

Kategorisierung	Beschreibung
Geräteart	Aufteilung nach Art bzw. Anwendungsbereich des Gerätes z.B. Lichtsteuerung, Sicherheit/Zugangskontrolle, Unterhaltung und Kontrolle, smarte Küche, HLK Kontrolle, smarte Möbel, häusliches Gesundheitswesen, Haushaltsgeräte
Service (Balta, 2013)	Aufteilung in drei große miteinander verknüpfte Gruppen Sicherheit, Energieverbrauch und -management sowie Lifestyle-Unterstützung. Deren Anwendungen enthalten Schnittmengen, die sich in die Untergruppen Energieeffizienz, Kommunikation, Unterhaltung, Gesundheitswesen, betreutes Wohnen, Komfort und Sicherheit gliedern
Segment (GSMA, 2012)	Aufteilung in Konnektivität, Serviceermöglichung sowie Geräte und Services, die dann nach Unterhaltung, Haussicherheit, Betriebsmittel bzw. Versorgung und mobiler Gesundheit weiter spezifiziert sind
Technologie (GSMA, 2012)	Aufteilung nach Netzwerkstandard bzw. Industrieinitiative (z.B. ZigBee, Wi-Fi, Bluetooth, IEEE 1905 oder UPnP), die dann weiter nach Konsument, Smart Energy, Gesundheit, intelligente Verbrauchsmessung, Unterhaltung oder einem allgemeinen Verwendungszweck klassifiziert werden

Eine Kategorisierung smarter Produkte kann also nach Geräteart, Art des angebotenen Service, nach Marktsegment oder nach eingesetzter Technologie erfolgen. Viele Produkte stellen allerdings auch geräteartübergreifende Funktionen zur Verfügung (z.B. DCH G020 mydlink Home Connected Home Hub mit eigenem Protokoll "D-Link's HNAP protocol"). Hier ist besonders die serviceorientierte Kategorisierung mit ihren Untergruppen als Schnittmengen übergeordneter Kategorien geeignet. Mit dem Hintergrund des signifikanten gestalterischen Einflusses der Marktteilnehmer, die die technischen Schnittstellen und AI-Anwendungen (Amazon, Google, Apple, Microsoft, etc.) für SmartHome-Geräte bereitstellen, ist eine Aufteilung nach Balta (Balta 2015) ebenfalls empfehlenswert.

In der Abbildung 15: SmartHome-Ökosystem wird eine Darstellung und Kategorisierung des SmartHome-Ökosystems vorgeschlagen, die versucht die Kategorien, Servicearten, Technologien und einen Auszug einiger wesentlicher Marktteilnehmer und Anwendungsbeispiele zusammenzuführen. Diese Darstellung hat keinen Anspruch auf Vollständigkeit und soll lediglich einen Überblick über den SmartHome-Markt mit seiner technischen Komplexität liefern.

Abbildung 15: SmartHome-Ökosystem



Quelle: Eigene Darstellung in Anlehnung an GSMA, Statista, Balta

In dieser Darstellung sind jedoch keine Organisationen dargestellt, die steuernd und regelnd Einfluss auf den SmartHome-Markt nehmen. Diese sind in ihrer Rolle und Verantwortung aber besonders herauszustellen. Zu nennen sind hier die Regulierungsbehörden, die Organisationen zur Entwicklung von Standards, Industrieverbände, die Datenschutzgesetze und Datensicherheitsanforderungen, die alle eine signifikante Steuerwirkung aufweisen.

Die Unternehmen des SmartHome-Marktes können ebenfalls klassifiziert werden. Eine mögliche Einteilung kann anhand ihrer angebotenen und in SmartHome-Produkten verbauten Komponenten erfolgen:

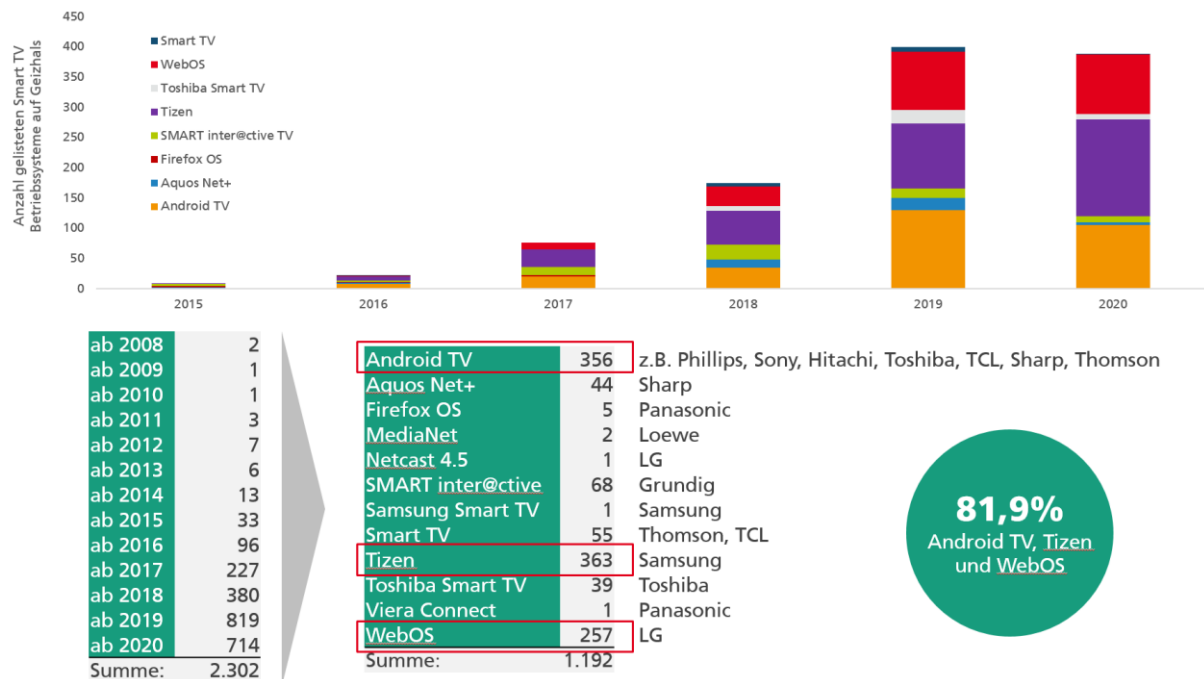
- ▶ Module (z.B. tyco, Lutron, OSRAM, Philips, Honeywell, Siemens, Schneider Electric)
- ▶ Sensoren (z.B. Bosch, FLIR, SENSIRION, ams, STMicroelectronics)
- ▶ Gateways (z.B. Amazon, Samsung, Microsoft, Google, Apple)

Unternehmen in der Modulentwicklung sind besonders gut im SmartHome-Markt positioniert, da sie gerade im Bereich Heizung, Lüftung, Klimatechnik und in der Lichtsteuerung bereits beim Bau von Gebäude involviert werden. Sensorhersteller und deren Zulieferer (z.B. SK Hynix) werden keinen signifikanten Einfluss auf den SmartHome-Markt haben, sind aber als Partner erforderlich. Gateway-Entwickler wie Google, Amazon, Apple, Microsoft und Samsung gestalten den Markt über ihre Interfaces und AI-Anwendungen und haben ebenfalls starke gestalterische Kraft auf die aktuelle und zukünftige Marktentwicklung. An dieser Stelle ist allerdings anzumerken, dass diese scharfe Trennung nicht in jedem Fall möglich ist, da sich einige Unternehmen auch klassifizierungsübergreifend positionieren. (siehe Abschnitt 4.4.4)

4.4.2 Untersuchung des SmartTV-Segments

Zur Untersuchung des SmartTV-Segments wurden ca. 2.300 Datensätze von SmartTVs und deren Betriebssysteme ausgewertet (Abbildung 16). Die Datensätze stammen von der Vergleichsplattform „Geizhals“, die seit 2008 die Klassifizierung „SmartTV“ bei Fernsehgeräten listet und deren Anzahl seit dem Jahr 2015 stark zugenommen hat. Diese Datensätze dienen der Auswertung der Marktteilnehmer und der eingesetzten Betriebssysteme, da deren Updatezyklen, eingesetzte Frameworks, kompatible Zusatzgeräte und der durch App-Stores bereitgestellte Funktionalität mögliche Obsoleszenzrisiken beeinflussen.

Abbildung 16: SmartTVs und deren Betriebssysteme 2008 bis 2020



Quelle: Eigene Darstellung

Bei den ausgewerteten Daten handelt es sich lediglich um die zum Verkauf stehenden Geräte und repräsentieren damit keine Absatzzahlen. Sie bieten jedoch eine gute Orientierung zur Identifikation relevanter Hersteller und deren technischer Merkmale.

Wurden im Jahr 2008 lediglich zwei Fernsehgeräte mit der Bezeichnung „SmartTV“ auf der Geizhals-Plattform gelistet, sind es im Jahr 2020 bereits 714 verschiedene Geräte, die neu zum Verkauf angeboten werden. Wann bestimmte Geräte nicht mehr zum Verkauf standen ist leider aus den Daten nicht ersichtlich. Das Wachstum in der Geräteanzahl und -vielfalt ist bis 2019 aber gut erkennbar. Bei 1.192 der 2.302 Geräte sind die darauf laufenden Betriebssysteme angegeben. Hier sind Samsung und LG mit ihren eigenen Betriebssystemen Tizen und WebOS als größte Hersteller im SmartTV-Markt eindeutig zu identifizieren. Android TV wird hingegen von vielen Herstellern wie z.B. Philips, Sony, Hitachi, Toshiba, TCL, Sharp und Thomsen eingesetzt. Insgesamt ist Android TV, Tizen und WebOS auf 81,9% aller angebotenen Geräte installiert. Im Jahr 2020 sind die wesentlichen Marktteilnehmer im deutschsprachigen Raum: Samsung mit 181, LG mit 103, Philips mit 56, Sony mit 55 und Panasonic mit 56 verschiedenen angebotenen Geräten. An dieser Stelle ist noch einmal anzumerken, dass es sich lediglich um die Anzahl der gelisteten Geräte handelt. Die Gewichtung ist also verzerrt, sollte ein Hersteller viele unterschiedliche Geräte zum Verkauf anbieten. Es ist nicht ersichtlich, ob ein Hersteller, der ggf.

nur einige wenige verschiedene Modelle anbietet, trotzdem hohe Verkaufszahlen vorweisen kann.

Dennoch sind besonders die dominierenden Betriebssysteme AndroidTV, Tizen und WebOS herauszustellen. Die größte Gemeinsamkeit dieser drei Systeme liegt in ihrer Kompatibilität zu anderen Geräten: Smartphones, Smart Devices wie Kühlschränke und Wearables, sowie Smartwatches werden unterstützt (zusätzlich Kameras bei Samsung). Die folgende Übersicht zeigt verschiedene Parameter und Eigenschaften der Betriebssysteme, die generell als obsoleszenzrelevante Faktoren in den Wirkungskategorien Funktionalität, Kompatibilität und Sicherheit angesehen werden können:

Tabelle 11: Eigenschaften von Smart-TV-Betriebssystemen

	Tizen	WebOS	Android TV
Betriebssystem-Hersteller	Samsung	LG Electronics	Google
Kernel	Linux	Linux	Linux
Empfohlenes App-Framework	Tizen .NET	Web App (HTML5) Node.js-Dienste	Java, Kotlin (wie Android Smartphones)
Aktuelle Major Version	5.5 / 6.0	3 / 4 / 5 / 6	11
Funktionalität			
Apps aus anderen Quellen installierbar	Ja	Nein	Teilweise / Umwege
Web-Apps	Ja	Ja	Ja
App Store	Samsung Apps TV	LG content store	Google Play Store herstellereigene Stores
Anzahl Apps im Store	Ca. 300	Ca. 700	Ca. 6500 Vorteil: Android-Smartphone-Apps können leicht adaptiert werden
Spiele Google Stadia	Nein	Manche ab 2021	Aktuelle Modelle
Kompatibilität			
Smartphone-Steuerung	Ja iOS/Android	Ja iOS/Android	Ja iOS/Android
Sprachassistenten	Bixby Alexa Google Assistant	LG ThinQ AI Alexa Google Assistant	Google Assistant Alexa
Smart-Home-Einbindung	Samsung Smartthings Homebridge: Apple Home Kit	QuickSet Apple Home Kit	Google Home Homebridge: Apple Home Kit

	Tizen	WebOS	Android TV
Sicherheit			
Versionsvielfalt	Version 1.0 – 2012 Version 2.0 – 2013 Version 3.0 – 2017 Version 4.0 – 2017 Version 5.0 – 2018 Version 6.0 – 2020	Geräte mit 3.x, 4.x und 5.x erhältlich, 6.0 angekündigt (1/2021)	Release-Timeline parallel zu Android für Smartphones, Hersteller müssen Updates selbst integrieren
Update-Verfahren	Über das OS	Über OS, alternativ USB	Über das OS
Typische Updateperiode	Major-Updates teilweise auch für ältere Modelle, teilweise Funktions-Updates auch für ältere Modelle	Neue Major-Versionen nur für neue Modelle, keine Aussage zu Sicherheitsupdate-Zyklen gefunden	Vereinbart: bis 2019 drei jetzt zwei Major-Versionen / Jahre: Achtung: HW-Voraussetzungen
Bug-Bounty Programm	Ja	Nein	Ja
Berichte zu Security-Problemen	Regelmäßig	Unregelmäßig	Unregelmäßig

AndroidTV, Tizen und WebOS verzeichnen Gemeinsamkeiten beim verwendeten Kernel und den Kompatibilitäten zu Services und Geräten anderer Hersteller und Anbieter. Bei der Update-Politik, den verwendeten Frameworks und der Anzahl der verfügbaren Anwendungen der jeweiligen AppStores unterscheiden sich die drei Betriebssysteme voneinander.

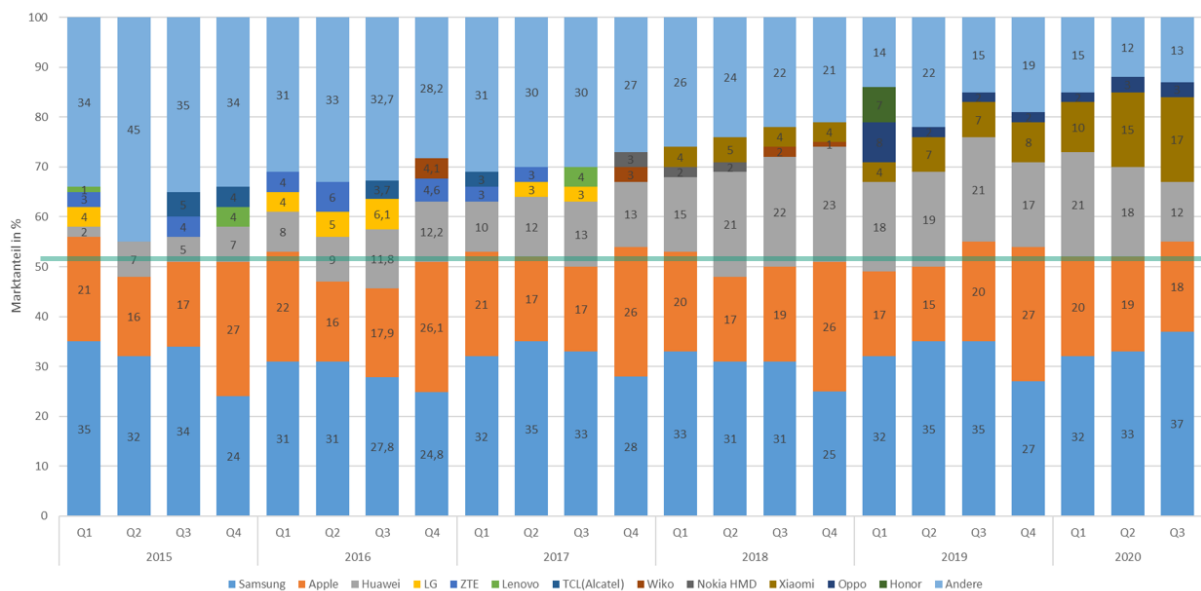
Im Januar 2021 kündigte LG an 20 weitere SmartTV-Hersteller für WebOS lizenzieren zu wollen. Namentlich genannt wurden RCA (primär US-Markt; nicht Deutschland), Konak (primär US-Markt; nicht Deutschland) und Ayonz (Marke Blaupunkt; primär Australien). Aktuell wird sich die Lizenzierung nur auf WebOS in der Version 5.0 beschränken. LG wird ihre eigenen neuen Geräte aber bereits mit WebOS 6.0 anbieten. Und nach aktuellem Kenntnisstand wird es kein Update von WebOS 5.0 auf Version 6.0 für bereits verkaufte LG-Geräte oder Geräte der Lizenznehmer geben. Samsung hat ebenfalls angekündigt Tizen zu lizenzieren und damit den Zugang auf ihr Betriebssystem auch anderen Herstellern zu ermöglichen.

4.4.3 Untersuchung des Smartphone-Segments

In der ersten Jahreshälfte 2019 kauften deutsche Verbraucher ca. 9,4 Mio. Smartphones. Diese Absatzzahlen gingen im Vorjahreszeitraum um 7% zurück (Statista 2019). Dieser Abwärtstrend zeigt sich seit dem Jahr 2015 in dem der höchste Wert von 26,2 Mio. verkauften Smartphones in Deutschland erreicht wurde (Statista 2019). Der Smartphone-Markt scheint in Deutschland gesättigt zu sein.

In Abbildung 17 sind die Marktanteile verschiedener Smartphone-Hersteller in Europa vom Jahr 2015 bis 2020 aufgetragen. Hier sind zwei wichtige Entwicklungen zu beobachten.

Abbildung 17: Marktanteile Smartphone-Hersteller in Europa von 2015 bis 2020



Quelle: Statista 2020

Samsung und Apple haben zusammen seit 2015 einen fast konstanten Marktanteil von 51% im gesamten europäischen Smartphone-Markt (grüne horizontale Linie). Samsungs Marktanteil liegt in diesen Fünfjahreszeitraum durchschnittlich bei ca 31% und Apples bei ca. 20%. Aber innerhalb eines Jahres schwanken die Anteile dieser beiden Hersteller. Im 4. Quartal jedes Jahres steigt Apples Marktanteil und übertrifft den von Samsung sogar in den Jahren 2015, 2016 und 2018. Diese Volatilität ist wahrscheinlich auf die Veröffentlichungszeitpunkte neuer Geräte der beiden Hersteller zurückzuführen.

Die zweite Erkenntnis, die aus Abbildung 17 gewonnen werden kann, ist der Konkurrenzkampf der anderen Hersteller, die sich die übrigen 49% des Marktes aufteilen. Der Bereich „Andere“ sinkt stark seit 2015 und wird von den chinesischen Herstellern Huawei, Xiaomi und Oppo übernommen. BBK Electronics (Oppo, Vivo, OnePlus, Realme) sind nach Stückzahl (nach Samsung und vor Apple) zum zweitgrößten Hersteller von Smartphones weltweit aufgestiegen. Der Marktanteil von Huawei ist von 2% im ersten Quartal 2015 auf 21% im ersten Quartal 2020 gewachsen. Xiaomi ist von 4% im ersten Quartal 2019 auf 17% im dritten Quartal 2020 gewachsen, womit das Unternehmen in Europa leicht hinter Apple liegt. Der weitere Ausbau der Marktanteile chinesischer Hersteller ist in den letzten Jahren deutlich sichtbar. In Europa lag die Smartphone-Penetration im Jahr 2018 bei 77,6% und verzeichnet damit ein noch wesentliches Marktpotential für die Smartphone-Hersteller dieser Welt.

4.4.4 Systeme und Plattformen des SmartHome-Marktes

Das Bundesministerium für Wirtschaft und Energie (BMWi) hat 2016 die Studie „SmartHome2Market“ veröffentlicht, die die Marktperspektiven sowie die Systeme und Plattformen im Bereich SmartHome diskutieren. Diese Studie gibt eine sehr gute Übersicht über die Systeme und Plattformen. Sie dient als Basis zur Ableitung möglicher Indikatoren und damit einem Vorschlag zur Bewertung obsoleszenzbedingter Risiken für die Lebenszeitverkürzung durch Software immer in Bezug auf Sicherheit, Kompatibilität und Funktion.

Wichtig an dieser Stelle ist zu erwähnen, dass zur Vereinfachung in den folgenden Ausführungen die technischen Lösungen und Plattformen nicht getrennt voneinander betrachtet werden.

Manche Plattformen nutzen technische Systeme, die hier separat aufgelistet sind. Das liegt vor allem an den engen Verflechtungen zwischen den Akteuren und der Heterogenität der Lösungen (siehe Abbildung 19). Die in der folgenden Übersicht definierten Indikatoren sind Vorschläge und haben keinen Anspruch auf Vollständigkeit. Sie dienen als Orientierungshilfe und beziehen sich auf die genannten Wirkungskategorien, um risikoreichere von risikoärmeren Lösungen in Bezug auf softwarebedingte Obsoleszenz aufzuzeigen.

Tabelle 12: Indikatoren für das Risiko softwarebedingter Obsoleszenz von SmartHome-Systemen

Indikator	Erläuterung	Skala
Anzahl der (assoziierten) Partner	Kleinster Wert ist hier 0 und wird damit als „keine“ bewertet (z.B. bei Fritz!Box); Der höchste vorliegende Wert mit über 60.000 bei KNX wird als „Hoch“ bewertet	Keine Niedrig Mittel Hoch
Produkt vollumfänglich am Markt platziert	Wichtig zur Einordnung, ob bereits ein Markteintritt (Early Market) stattfand; Nein: Produkt noch Forschungs idee bzw. in der technischen Entwicklung befindlich (Forschungsinitiativen) Ja: Produkt am Markt für die Endnutzer*innen erhältlich	Ja / Nein
Installation und Steuerung komplett durch Nutzer*innen realisierbar (Plug&Play)	Endnutzer*in kann die Installation und Steuerung alleine realisieren und muss nicht durch Dritte durchgeführt werden; Einschränkung: Manche Lösungen sind B2B-orientiert und nicht an Endnutzer*innen gerichtet. Hier gibt es also eine Differenzierung bedingt durch das Geschäftsmodell. Das Projekt SobO konzentriert sich auf den Endnutzermarkt. Allerdings soll die Übersicht ein möglichst vollständiges Bild der Marktakteure zeigen und listet auch B2B-Anbieter. Die Bewertung erfolgt aber aus Privatnutzersicht. Ja: Plug&Play Nein: kein Plug&Play oder anderes Geschäftsmodell	Ja / Nein
Offene Nutzung der Lizenz	Ja: Open Source Nein: „closed shop“, Lizenzierung, ganz oder teilweise proprietär, nur bedingte Nutzung	Ja / Nein
Integration weiterer Hersteller oder Dienstleister	Varianz der Partner; Also z.B. zusätzliche Integration von IT-Herstellern, Anwendungsherstellern, Sensor-Aktor-Herstellern, Handwerker/Dienstleistern oder Communities	Keine Niedrig Mittel Hoch
Anzahl kommerziell erhältlicher kategorie-übergreifender Lösungen	Wie variantenreich ist die Anwendung in Bezug auf die Kategorien Gebäudesicherheit, Energy Management, Home Entertainment, Vernetzung und Steuerung, Komfort und Licht	Keine Niedrig Mittel Hoch
Anwendung Cloud-basierter Dienste	Ja: Cloud-Anwendung Nein: keine Cloud-Anwendung Wichtig, falls ein Dienst eingestellt wird und das Produkt keine offenen Nutzungsrechte des Quellcodes hat, kann das Produkt gar nicht oder nur noch eingeschränkt weiter genutzt werden	Ja / Nein

Abbildung 18: Bewertungsmatrix von SmartHome-Anbietern

Name des Systems bzw. der Allianz (Treiber)	Anzahl der (assoziierten) Partner	Produkt vollumfänglich am Markt platziert	Installation und Steuerung komplett durch Nutzer realisierbar (Plug&Play)	Offene Nutzung der Lizenz	Integration weiterer Hersteller oder Dienstleister	Anzahl kommerziell erhältlicher kategorieübergreifender Lösungen	Anwendung Cloud-basierter Dienste	Bewertung
AllSeen Alliance (Qualcomm)	Mittel	Nein	Nein	Ja	Niedrig	Niedrig	Nein	5
ECHONET LITE	Mittel	Nein	Nein	Nein	Niedrig	Mittel	Nein	5
EEBus Initiative e. V.	Mittel	Nein	Nein	Nein	Niedrig	Niedrig	Nein	4
Fritz!Box (AVM)	Keine	Ja	Nein	Nein	Niedrig	Mittel	Ja	5
Home Gateway Initiative	Niedrig	Nein	Nein	Nein	Niedrig	Niedrig	Nein	3
Home Kit (Apple)	Niedrig	Nein	Ja	Nein	Niedrig	Niedrig	Ja	5
HyperCat (British Standards Institution)	Mittel	Nein	Nein	Nein	Niedrig	-	Nein	3
Open Connectivity Foundation	Mittel	Nein	Nein	Ja	Niedrig	Niedrig	Nein	5
openHAB	Keine	Nein	Nein	Ja	Niedrig	Hoch	Ja	6
QIVICON (Deutsche Telekom)	Niedrig	Ja	Nein	Nein	Niedrig	Mittel	Ja	6
Belkin WEMO	Keine	Ja	Ja	Nein	Mittel	Mittel	Nein	6
Bluetooth Low Energy (Bluetooth Smart)	Hoch	Ja	Nein	Ja	Mittel	Hoch	Nein	10
devolo	Keine	Ja	Ja	Ja	Niedrig	Mittel	Nein	6
digitalSTROM-Allianz	Niedrig	Nein	Nein	Nein	Niedrig	Niedrig	Nein	3
EnoCEAN Alliance	Mittel	Ja	Nein	Nein	Mittel	Mittel	Nein	7
Gigaset elements	Keine	Ja	Ja	Nein	Niedrig	Mittel	Nein	5
HomeMatic (IP)	Niedrig	Ja	Nein	Nein	Niedrig	Mittel	Nein	5
KNX (KNX Association)	Hoch	Ja	Nein	Ja	Hoch	Hoch	Nein	11
LON (LonMark International, LMI)	Mittel	Nein	Nein	Nein	Mittel	Mittel	Nein	6
Loxone Electronics GmbH	Niedrig	Ja	Ja	Nein	Niedrig	Mittel	Ja	7
RWE SmartHome	Niedrig	Ja	Ja	Nein	Niedrig	Mittel	Ja	7
SAMSUNG Smart Home	Keine	Nein	Nein	Nein	Niedrig	Niedrig	Ja	3
The Thread Group (Google/Nest)	Mittel	Nein	Ja	Nein	Niedrig	Mittel	Ja	7
TRON Smart Home	Mittel	Nein	Nein	Nein	Niedrig	Niedrig	Nein	4
Works with Nest (Google/Nest, Revolv)	Mittel	Nein	Ja	Ja	Mittel	Mittel	Ja	9
ZigBee	Mittel	Ja	Nein	Ja	Mittel	Hoch	Nein	9
Z-Wave	Mittel	Ja	Nein	Ja	Niedrig	Hoch	Nein	8

Quelle: Eigene Darstellung in Anlehnung an BMWi 2016

Die Punktzahl in der Bewertungsspalte ergibt sich aus der Summe der einzelnen Indikatoren, wobei „Keine“ dem Wert 0, „Niedrig“ dem Wert 1, „Mittel“ dem Wert 2 und „Hoch“ dem Wert 3 entspricht. Es wurde keine Gewichtung der Indikatoren vorgenommen. Diese sollte bei einer Weiterentwicklung und möglichen Erweiterung des Bewertungsmodells in Betracht gezogen werden. Das beste Ergebnis erreicht KNX mit einer Gesamtpunktzahl von 11. Das niedrigste Ergebnis von 3 und damit ein erhöhtes softwarebedingtes Obsoleszenzrisiko haben die Home Gateway Initiative (gibt es nicht mehr), Hypercat (gibt es so nicht mehr und verweist jetzt auf die British Standards Institution -BSI), die digitalStrom Allianz (Ist jetzt eine AG, die direkt Produkte mit Installationspartner vertreibt) und Samsung SmartHome. Hingegen gut aufgestellt sind Bluetooth Low Energy, KNX, Google Nest, Zigbee und Z-Wave. An dieser Stelle ist anzumerken, das Bluetooth Low Energy seit Version 5 vollständig im normalen Bluetooth Standard integriert. Die Liste der Systeme/Plattformen stammt aus dem Jahr 2016 und ist teilweise nicht mehr aktuell. Die Auswertung konnte jedoch zeigen, dass zwei der vier am geringsten bewerteten Lösungen sich nicht langfristig am Markt positionieren konnten. digitalSTROM und Samsung Smart Home haben sich hingegen am Markt etabliert, bedienen allerdings auch mit ihren unterschiedlichen Lösungsansätzen auch verschiedene Kundensegmente.

Wie bereits eingangs erwähnt, sind die Entstehungsgeschichten, technischen Umsetzungen, Initiativen und Geschäftsmodelle im SmartHome-Markt sehr variantenreich. Aus diesem Grund werden nun vier der Systeme/Plattformen kurz näher vorgestellt:

Die digitalSTROM-Allianz entstand in Zusammenarbeit der Forschungseinrichtungen Fraunhofer FOKUS, der Hochschule Luzern, der Universität Stuttgart, der Berner Fachhochschule und der TU Berlin. Zusammen mit den Unternehmen BNG Gebäudetechnik, NetSolar EBERLE und der BS2 AG sowie der ETH Zürich als Entwickler bildeten sie die digitalSTROM-Allianz. Das Angebot konzentrierte sich auf intelligente (Lüster-) Klemmen bzw. Zwischenstecker, die über Powerline eine Vernetzung vieler Geräte im Heim ermöglichen (ausgenommen sind aber z. B. Thermostate an Heizkörpern).

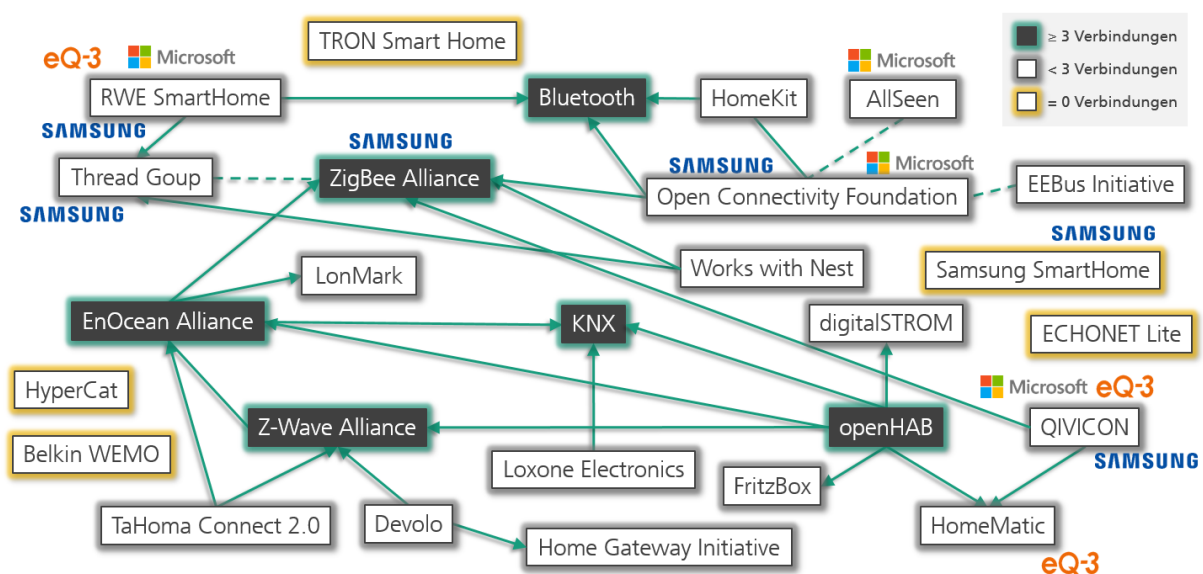
Samsung Smart Home hingegen ist eine unternehmenseigene Entwicklung, die jedoch stark von der 2014 gekauften SmartThings-Lösung geprägt ist. Samsung hat sich hier die Kompetenz und vor allem die Kompatibilität zu Geräten von z.B. Philips, Google Nest, Honeywell, ring Boser oder belkin eingekauft. Sie bieten ein IP-basiertes proprietäres System am Markt in Südkorea, Europa und den USA an, das mit Hilfe der SmartThings-App über 1.000 Geräte ansprechen kann.

Ein wieder anderes Entstehungsmodell weist die KNX Association auf. Sie ist eine Nachfolgeorganisation der European Installation Bus Association (EIBA) mit den Gründungsmitgliedern aus unterschiedlichen Bereichen. Hier sind z.B. Siemens, Busch-Jaeger, JUNG, Wieland und die Hager Group zu nennen. Sie entwickelten den Nachfolgebus aus dem Europäischen Installationsbus (EIB), dem European Home Systems (EHS) Protokoll und dem BatiBus. Die früher als „Konnex“ bekannte Vereinigung hat heute 500 Mitglieder in 190 Ländern und umfasst 8.000 Produkte, die primär durch Fachkräfte installiert werden. Die Anbindung wird über Twisted Pair, Funk, Powerline und IP realisiert.

Als letztes soll an dieser Stelle Google Nest kurz vorgestellt werden, die eine ähnliche Strategie wie Samsung verfolgt. Hinter Google steht heute der Mutterkonzern Alphabet, der ebenfalls im Jahr 2014 das Unternehmen Nest und gleichzeitig revolv und dropcam kaufte. Mit dieser Übernahme wurde das Know-how und die Kompatibilität zu Produkten der Unternehmen Nvidia, GE, ring, logitech, belkin, Lenovo, WiZ, LG und Philips in das Produktportfolio überführt. Die Kommunikation von Google Nest-Geräten erfolgt über WiFi oder ZigBee. „Works with Nest“ wurde kurze Zeit später zu „Works with Google Assistant“. Wird also ein bestehender Nest-Account nicht zu einem Google-Account migriert, werden dem Nutzer*innen lediglich noch Sicherheitsupdates zur Verfügung gestellt (Stand 2019). Auch neue Geräte z.B. Philips Hue können nicht mehr mit einem alten Nest-Account verbunden werden und die Kompatibilität zu Amazons Alexa ist aktuell noch nicht wieder möglich.

Gerade anhand des Verhaltens von Samsung und Google zeichnet sich die wesentliche Rolle der Schnittstellen zwischen den Lösungen verschiedener Anbieter zur Etablierung auf dem Smart-Home-Markt ab. In Abbildung 19 wurden die in Abbildung 18 aufgeführten Systeme/Plattformen auf genau diese Kompatibilität untersucht.

Abbildung 19: Beziehungsdiagramm SmartHome-Anbieter



Quelle: Eigene Darstellung

Das Beziehungsdiagramm zeigt Kompatibilität (Pfeil), geplante Kompatibilität/Schnittstellen (gestrichelter Pfeil) und keine Kompatibilität (keine Verbindung) zwischen den verschiedenen Systemen/Plattformen. Die grün hinterlegten Anbieter haben drei oder mehr Verbindungen und die weiß hinterlegten Anbieter weniger als drei Verbindungen. KNX, Bluetooth, ZigBee, EnOcean, Z-Wave und openHAB bieten demnach die meisten Schnittstellen zu anderen Lösungen. Gelb gekennzeichnete Anbieter, wie z.B. Samsung SmartHome, TRON Smart Home oder ECHONET (nur in Japan) wiesen keine Verbindungen zum Untersuchungszeitpunkt auf. Das Beziehungsdiagramm zeigt noch einmal deutlich, wie eng die unterschiedlichen Lösungsansätze doch miteinander verzahnt sind. Gleichzeitig ist bei der Untersuchung aufgefallen, dass drei Unternehmen besonders viel Einfluss auf den SmartHome-Markt nehmen. Wie in Abbildung 19 zu sehen, partizipieren Samsung, Microsoft und eQ-3 an mehreren Systemen und bieten teilweise zusätzlich eigene Lösungen an.

4.5 Zusammenfassung und Fazit

Für die bessere Einschätzung der künftigen Entwicklung und umweltpolitischen Relevanz wurde im Rahmen der Studien eine Untersuchung der Marktentwicklung von Haushaltsgeräten bis zum Jahr 2030 vorgenommen. Im Fokus standen sogenannte SmartHome-Produktsysteme, wobei mit Smartphones und Smart TVs zwei besonders dynamische und aufgrund ihrer hohen Verbreitung relevante Produktsegmente vertiefend betrachtet wurden.

Die Untersuchung der Haushaltsentwicklung in Deutschland zeigt bis zum Jahr 2030 ein Marktpotential von 837.000 neuen Ein- und Zweipersonenhaushalten für den SmartHome-Markt. Grundsätzlich lassen sich die Marktteilnehmer in Modulentwickler, Sensorhersteller und Gateway-Entwickler einteilen. In der Modulentwicklung am stärksten positioniert sind Unternehmen wie Phillips, OSRAM, SIEMENS oder Schneider Electric. Zudem nehmen Google, Amazon, Apple und Microsoft durch ihre Anwendungen im Gateway-Bereich eine dominante Rolle für die aktuellen und zukünftigen Marktentwicklungen ein. Für die betrachteten Produktgruppen lässt sich im Einzelnen festhalten, dass im Segment der SmartTVs die Betriebssysteme AndroidTV, Tizen und WebOS mit einem Anteil von 81,9% der zum Verkauf stehenden Geräte im deutschsprachigen Raum als wesentliche technologische Treiber gelten. Im

Smartphone-Segment besetzten die vier Hersteller Samsung, Apple, Huawei und Xiaomi 84% des europäischen Marktes. Im Vergleich zum deutschen Smartphone-Markt, der bereits erste Anzeichen einer Sättigung aufweist, lag die Penetrationsrate auf dem europäischen Markt im Jahr 2018 bei 77,6% (Statista 2021).

Für die abschließende Einschätzung wird ein Vorschlag für ein Bewertungssystem mit Hilfe von sieben Indikatoren vorgestellt, welches eine grobe Abschätzung softwarebedingter Obsolenzrisiken von Systemen und Plattformen ermöglicht. Das Bewertungssystem orientiert sich an den bereits im Vorfeld identifizierten Wirkungskategorien und Anforderungen an die Kompatibilität, Funktionalität und Sicherheit von Software.

5 Analyse empirischer Fälle von softwarebedingter Obsoleszenz

Im Bereich der Consumer-Elektronik (Haushaltsgeräte, IKT, Unterhaltungselektronik) gibt es bisher nur wenige dezidierte Studien und Arbeiten zum Thema der softwarebedingten Obsoleszenz (Poppe et al. 2021; Wagner et al. 2020). Dennoch wird auf die Problematik direkt und indirekt in fast allen neueren Arbeiten im Kontext von „geplanter Obsoleszenz“ (Poppe & Longmuß 2019; Hess 2018; Prakash et al. 2016; Schridde & Kreiß 2013) oder nachhaltiger Software und Green IT hingewiesen (Gröger & Köhn 2015; Lohmann 2015; Hilty et al. 2015). Zudem wird das Thema im Rahmen von umweltpolitischen Maßnahmen vereinzelt adressiert (siehe Pkt. 8.2) und es gab bereits erste rechtliche Schritte in anderen Ländern (siehe Pkt. 8.2.5). Aufgrund fehlender Übersichtsliteratur wird im Folgenden anhand ausgewählter Fallbeispiele und Studien auf grundsätzliche Entwicklungen und Problematiken im Bereich der Consumer-Elektronik aufmerksam gemacht sowie weitere Forschungsbedarfe identifiziert. Die Beschreibung erfolgt überwiegend qualitativ und explorativ. Im Vordergrund der Analyse steht hierbei zum einen der grundsätzliche Wirkmechanismus der zur Obsoleszenz des Produkts beitragen kann sowie das Rechtfertigungsverhalten der Hersteller und eine rechtliche Einschätzung des Falls.

Die qualitative Fallanalyse wird ergänzt durch eine vertiefende systematische Untersuchung. Hierzu wurden weitere aktuelle Fälle ausgewählt, welche sich in Produktkategorie sowie Obsoleszenzursache und -auswirkung unterscheiden. Die Analysemethode wurde auf Basis der wenigen existierenden theoretischen Ansätze zu einem Analyse-Framework adaptiert, welches sich auf Consumer-Elektronik anwenden lässt. Folgende Tabelle zeigt eine Auswahl von Fallbeispielen mit differenzierten Ursachen.

Tabelle 13: Ausgewählte Fallbeispiele zur softwarebedingten Obsoleszenz bei Konsumgütern

Fallbeispiel	Hersteller	Beschreibung	Bewertung
Zählvorrichtung und Schutzvorrichtung bei Druckern	Epson	Einige Druckerhersteller setzen Zählvorrichtungen in ihren Produkten ein, die ab einer bestimmten Anzahl gedruckter Seiten, die Funktion beschränken oder komplett einstellen.	Zählvorrichtungen beschränken nachweislich die Funktionsdauer, sollen laut Hersteller aber eine durchgehend hohe Druckqualität gewährleisten und sollen als Schutzvorrichtung dienen.
Trade-Up Programm mit Recyclingmodus bei Bluetooth Lautsprechern	Sonos	Sonos bot seinen Kund*innen Anfang 2020 an, alte Lautsprecher für den Kauf gegen neue in Zahlung zu nehmen („trade-up“). Hierfür sollten Nutzer*innen im Vorfeld einen Recyclingmodus aktivieren, welcher die Geräte in der Folge unbrauchbar macht.	Sonos hat das Programm nach öffentlicher Empörung eingestellt. Trade-Up oder Trade-In Programme werden jedoch nach wie vor von einer Vielzahl an Herstellern und Handel betrieben, wobei unklar ist, was mit den alten Geräten passiert.
Performance Degradierung zur Verlängerung der	Apple iPhone	Mit Hilfe eines Updates wurden bei älteren iPhones bestimmte	Apple informierte seine Nutzer*innen nicht hinreichend über die Folgen des Updates und wurde hierfür bereits in

Fallbeispiel	Hersteller	Beschreibung	Bewertung
Batterielebensdauer bei Smartphones		Softwarefunktionen eingeschränkt, um die Batterie zu schonen.	Frankreich und Italien verklagt (siehe Pkt. 8.2.5). Softwarebasiertes Batteriemanagement wird heute jedoch in einer Vielzahl von Produkten zur Performance- und Lebensdaueroptimierung eingesetzt.
Supportdauer für Softwareupdates bei Smartphones	Android	Android bietet ein offenes Betriebssystem für Smartphone Hersteller und wird nur über einen begrenzten Zeitraum mit Updates versorgt.	Die Supportdauer (Sicherheitsupdates) für Android beträgt derzeit im Schnitt 3 Jahre. Bisher ist unklar, in wie weit fehlende Updates Nutzer*innen tatsächlich zum vorzeitigen Umstieg auf neue und zur Obsoleszenz ihrer alten Geräte bewegen.
End-of-Life cloudbasierter Plattformen	Belkin Wemo-Netcams	Einstellung der cloudbasierten Plattform nach 7 Jahren führt zum Funktionsverlust von Überwachungskameras	Belkin informierte am Ende April 2020 seine Nutzer*innen über die Einstellung der cloudbasierten Softwareplattform für seine Wemo-Netcams. Die Wemo-Netcams werden damit funktionsuntauglich.

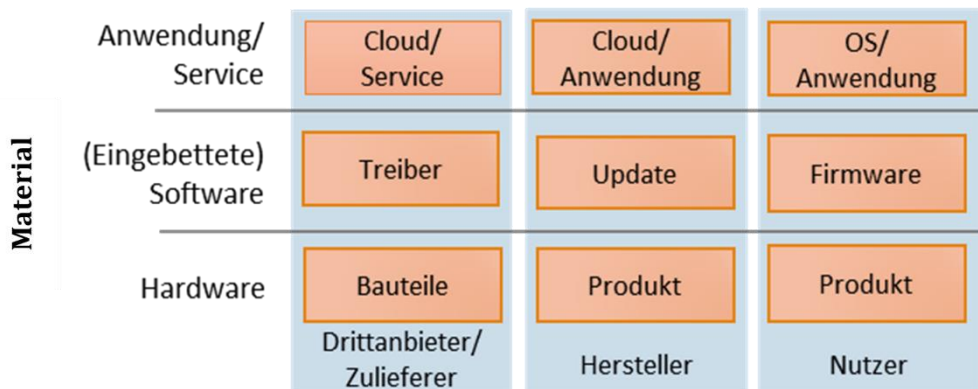
5.1 Kernaspekte und Ziele der Untersuchungen

Ziel der qualitativen Analyse ist es, Fallstudien von softwarebedingter Obsoleszenz zu rekonstruieren, um mögliche Ursachen-Wirkungsbeziehungen zu identifizieren und Handlungsempfehlungen abzuleiten. Für den Bereich industrieller Anwendungen existieren zwar bereits Ansätze zur grundsätzlichen Charakterisierung, diese eignen sich aber nur bedingt für die Analyse von Haushaltselektronik. Für die Erstellung eines Obsoleszenz-Profiles wurde methodisch ein zweistufiger Ansatz entwickelt. Das Profil ergibt sich aus einer Charakterisierung eines Falls hinsichtlich des Abhängigkeitspfads zwischen Software-Hardware (Material) sowie den beteiligten Akteuren: Hersteller – Nutzer*in (Milieu bzw. Domäne):

Charakteristika	Dimension	Analyse Ansatz
▶ Software - Hardwaresystem	→ Material	} 1. Systemdiagramm
▶ Hersteller - Benutzer Beziehung	→ Milieu	
▶ Ursachen, Effekte	→ Genese	→ 2. Ursache-Wirkung Model

Aufgrund der Diversität und Komplexität der Fälle wurde die zugrunde liegende Methodik iterativ überarbeitet. Eine detaillierte Beschreibung der Herleitung der explorativen Herangehensweise bzw. Methodik sowie die Datengrundlage finden sich in den Publikationen von Wagner et al. 2020 und Poppe et al. 2021.

Abbildung 20: Das Systemdiagramm in dem Dimensionen Milieu (horizontal) und Material (vertikal) aufgespannt; exemplarische Darstellung der Charakteristika in roten Kästen



Quelle: Eigene Darstellung

Vorstehende Abbildung zeigt die Matrix der beiden Dimensionen Material (vertikal) und Milieu (horizontal). Die Dimension „Material“ ist vom OSI-Modell (ITU, "7498-1:1994, ISO/IEC,") inspiriert, das hier auf drei Hauptschichten Hardware, eingebettete Software und Anwendung oder Service komprimiert ist, wobei die Schichten nicht zwangsläufig eine technische Verbindung aufweisen. Die Dimension „Milieu“ besteht aus Drittanbieter, Hersteller und Anwender. Wechselwirkungen und Abhängigkeiten zwischen Schichten und Dimensionen sind multidirektional. Die Produktkategorien (orange Kästen in der Abbildung) werden abgeleitet von den untersuchten Fallstudien (für die Produktsegmente Entertainment, Smarthome und Smartphone).

Das Ursache-Wirkungsmodell umfasst drei aufeinanderfolgende Schritte: (1) Ursachen, die meist zu (2) einer bestimmten Reaktion von Herstellenden führen, die (3) Softwareeffekte mit mehreren Effekten aufweist. Darüber hinaus werden Fallstudienspezifikationen (0) aufgezeichnet.

(0) Fallstudiengrenzen: Jeder Fall von Obsoleszenz wird durch das betroffene Modell, den Veröffentlichungspreis (Listenpreis), das Alter des Geräts bis zum Fall der Veralterung der Software sowie die Benachrichtigung über Softwareproduktänderungen (SPCN) berücksichtigt.

(1) Ursache: Zur Kategorisierung der Ursachen wurde der Abhängigkeitsbaum von Barthels et al. (2012) auf Cloud-Dienste erweitert, die der Kategorie „logistisch“ zugeordnet sind.

(2) Reaktion: Ursachen führen zu Reaktionen bzw. Maßnahmen seitens des Herstellers, wozu beispielsweise OTA-Updates (Over-the-Air), Cloud- und Support-Abbruch zählen. Die offizielle Rechtfertigung der Maßnahmen durch den Hersteller wurde kategorisiert nach Sicherheit, Qualität, Leistung (Systemgeschwindigkeit, Funktionalität) und Ressourcen (finanziell).

(3) Wirkung: Um die Auswirkungen der Maßnahmen auf die Softwarequalität zu gruppieren, wurden die Kategorien der ISO 25010 (Funktionalität, Leistung, Sicherheit, Wartbarkeit, Zuverlässigkeit, Portabilität, Kompatibilität, Benutzerfreundlichkeit) auf vier Merkmale eingegrenzt (LEUF-Schema, siehe Abbildung 1).

In den folgenden Kapiteln werden die Fälle zunächst qualitativ bewertet.

5.1.1 Fall 1: Zählvorrichtung bei Druckern

Ab Mitte der 2000er Jahre häuften sich Beschwerden von Nutzenden im Internet, das einige Drucker nach einer bestimmten Laufzeit und Nutzungsdauer nicht weiter funktionieren. Diese

Praxis wurde insbesondere bei Geräten des Herstellers EPSON beschrieben und diente auch für populärwissenschaftliche Arbeiten und Studien als beliebtes Beispiel für eine vermeintlich geplante (vorzeitige) Obsoleszenz durch die Hersteller (Dannoritzer 2010; Schridde & Kreiß 2013). Tatsächlich nutzen einige Geräte verschiedener Hersteller eine softwarebasierte Abschaltvorrichtung, die nach einer gewissen Anzahl an gedruckten Seiten aktiviert wird oder zumindest eine Warnung ausgibt. Begründet wird dies unter anderem von Herstellern wie Epson mit dem Hinweis, dass es sich hier um eine Schutzvorrichtung handelt, die zum einen größere Schäden am Produkt vermeiden und zum anderen langfristig eine gleichbleibende Druckqualität gewährleisten soll (Epson 2020; Heise Online 2008). Der Fall löste bei vielen Verbraucher*innen Empörung aus und insbesondere in Frankreich wird seit 2017 im Rahmen des Gesetzes zur Vermeidung geplanter Obsoleszenz (LOI n° 2015-992) (siehe Pkt. 8.2.5) staatlich gegen den Hersteller ermittelt.

Der Einbau von Vorrichtungen zum Schutz vor Übernutzung oder Fehlnutzung ist rechtlich schwer zu bewerten, da insbesondere bei ausreichender Informierung der Verbraucher*innen im Vorfeld (vorvertragliche Pflichten) nicht von einer Täuschungsabsicht ausgegangen werden kann. Es stellt sich dennoch die Frage, ob eine solche Geschäftspraxis allgemeine Verkehrssitten verletzt, da Käufer*innen für ihr Geld eine adäquate Gegenleistung erwarten dürfen (Hess 2018: 266-269). Hess stellt dazu in einer rechtlichen Untersuchung fest:

„Es kommt wiederum entscheidend auf die Motivation des Verwenders des Chips an. Hatte der Hersteller/Verkäufer den Vorsatz, den Drucker vorzeitig außer Betrieb zu setzen und gab es dafür keinen rechtfertigenden Grund (z. B. Geräteschutz), dürfte der Vorwurf der Sittenwidrigkeit gerechtfertigt sein. Bei leichteren Arten der „käuferbenachteiligenden Lebensdauerplanung“, insbesondere der fahrlässigen Fehleinschätzung bei der Produktgestaltung geht der Vorwurf eines Verstoßes gegen die guten Sitten zu weit. Eine praktische Schwierigkeit dürfte – neben der Begründung der Sittenwidrigkeit – der Nachweis des im Rahmen von § 826 BGB erforderlichen Schädigungsvorsatzes darstellen (Hess 2018: 304)“.

Unabhängig von der rechtlichen Bewertung verdeutlicht der Fall eine wichtige Erkenntnis: Hersteller begründen softwarebasierte Funktionseinschränkungen und Konfigurationen mit qualitätssichernden Maßnahmen – eine Praxis, die auch beim Batteriemangement in einigen Geräten wie Laptops und Smartphones vorzufinden ist. Weit verbreitet und speziell bei Druckern ist in diesem Kontext auch das Ausschließen von Dritten in der Wertschöpfungskette, in dem nur Teile und Verbrauchsartikel wie Patronen vom Originalhersteller mit dem Gerät betrieben werden können. Auch in diesem Fall gibt es bereits kartellrechtliche Sammelklagen gegen den Hersteller Epson in Frankreich seit 2017 und den USA seit 2019 (The Recycler 2019), weil wieder befüllte Patronen von Dritten in den Geräten durch eine Software erkannt werden und nicht funktionieren.

Ein bekanntes Beispiel hierzu findet sich auch unter dem Begriff „Error 53“ beim Hersteller Apple, welcher vermehrt bei iPhone 6 Modellen ab Anfang Februar 2016 nach einem Update des Betriebssystems auf Version iOS 9.2.1 auftrat. Apple blockierte das iPhone, sofern nicht autorisierte Händler und Reparaturbetriebe den Fingerabdrucksensor „Touch-ID“ austauschten. Eine Praxis, die auch bei anderen Herstellern verbreitet ist und unter dem Begriff der „Serialisierung“ bekannt ist. Apple rechtfertigte die Maßnahme zwar als Schutzvorrichtung, beseitigte die Sperrung im Anschluss jedoch zügig, innerhalb weniger Wochen, mit einem Update und entschuldigte sich öffentlich für den Fehler. Eine Sammelklage in den USA wurde deshalb von den Richtern zurückgewiesen (Schäfer 2016), war jedoch erfolgreich in Australien mit einer Strafe in Höhe von 6.6 Millionen US-Dollar (Bisset 2018).

In den hier genannten Fällen zeigt sich, dass die angemessene Anwendung von Schutzvorrichtungen durch Software bisher eine rechtliche Grauzone darstellt und in einigen Fällen den vorzeitigen Funktionsverlust von Hardware bewirken kann. Es gilt daher vertiefend zu klären, wie viel Schutz vor Übernutzung und Falschnutzung eine funktionsbegrenzende Intervention von Software überhaupt rechtfertigt und welchem Rahmen die Serialisierung von Komponenten insbesondere Eigenreparaturen oder Reparaturen durch unabhängige Dritte unnötig behindern.

5.1.2 Fall 2: Trade-Up Programm bei Bluetooth Lautsprechern

Ende 2019 kündigte der Hersteller Sonos für seine Bluetooth Lautsprecher ein „Trade-Up“ Programm ab. Nutzenden von älteren Geräten der gleichen Marke wurde die Möglichkeit geboten, durch eine softwaregesteuerte Aktivierung eines Recycling-Modus, ihre intakten Lautsprecher innerhalb von 21 Tagen unbrauchbar zu machen. Im Gegenzug hierfür wurde den Nutzenden ein Rabatt auf die Anschaffung eines Neugeräts gewährt. Bei den relevanten Gebrauchtgeräten handelte es sich um Modelle, die im Zeitraum von 2007 – 2011 verkauft wurden, für die noch immer ein intakter Gebrauchtmarkt besteht. Zudem berichten Medien, dass laut des Unternehmens 92 Prozent aller Geräte, die jemals verkauft wurden, noch immer in Gebrauch sind (Kohlick 2020).

Gegenüber den Medien rechtfertigte der Hersteller die Maßnahme vor allem damit, dass Neukunden*innen, die womöglich ein gebrauchtes älteres Modell des Geräts erworben hätten, davor bewahrt werden, nicht die „Sonos experience“ zu erhalten, die sie sich versprochen haben: „For those that choose to trade-up to new products, we felt that the most responsible action was not to reintroduce them to new customers that may not have the context of them as 10+ year old products, and that also may not be able to deliver the Sonos experience they expected“ (Interview by: Welch 2019). Neben dem Verbraucherschutz rechtfertigte der Hersteller das Anreizprogramm zudem als Umwelt- und Ressourcenschutzmaßnahme.

Das Vorgehen und die Rechtfertigung des Herstellers löste viel Kritik aus, sodass sich der Hersteller Anfang 2020 dazu entschlossen hat, das Programm einzustellen. Am „Trade-Up“ Programm wird zwar weiterhin festgehalten, die betroffenen Geräte werden aber nicht mehr durch eine Software deaktiviert (Herbig 2020b).

Bemerkenswert an diesem Fall ist, dass Hersteller und Kunden durch den Einsatz von Software einvernehmlich eine vorzeitige Obsoleszenz intakter Geräte und Hardware bewirken.¹⁰ Es handelt sich hierbei um einen offensichtlichen Fall der „Komplizenschaft“ (Jaeger-Erben 2019). Fernab der Empörung zeigt der Fall, dass es heute mit Einverständnis der Verbraucher*innen rechtlich möglich ist, völlig intakte Geräte zu zerstören und dafür einen wirtschaftlichen Anreiz zu bieten. Das umweltpolitische Problem in diesem Kontext ist demnach nicht der Modus und die Mittel, sondern die Tatsache, dass es strukturell überhaupt möglich ist. „Trade-Up/In“ oder „Alt gegen Neu“ Programme sind heute bei Herstellern und Handel weit verbreitet und werden z.T. auch staatlich gefördert (Abwrack- und Verschrottungsprämie). Mit der sogenannten Obhutspflicht wurde im Rahmen des Kreislaufwirtschaftsgesetz im Jahr 2020 eine Gesetzesregelung in Deutschland eingeführt, welche die Vernichtung von Retouren und nicht verkaufter Neuware erschwert (Paragraf 23 Absatz 1 Satz 3 KrWG), für intakte Gebrauchtwaren gibt es jedoch keine konkrete Regelung.

¹⁰ Es gab im Jahr 2012 eine Ankündigung des Herstellers RunCore, eine Festplatte herzustellen, die sich per Knopfdruck zum Schutz der Daten physisch selbst zerstören kann (Rother 2012). Es blieb jedoch bei einer Ankündigung und es handelte sich hierbei um eine elektromechanische Zerstörung.

Es besteht weiterer Prüfungsbedarf, ob eine vorzeitige und vermeidbare Vernichtung von intakter Gebrauchtware zumindest im Kern der Abfallhierarchie im Kreislaufwirtschaftsgesetz widerspricht (Paragraf 6 KrWG). Diese sieht vor, dass Gebrauchtwaren vor der Entsorgung auf eine Vorbereitung zur Wiederverwendung (VzWv) geprüft werden (Richtlinie 2008/98/EG).

5.1.3 Fall 3: Performance-Degradierung bei Smartphones

Nachdem Nutzende sich Ende 2017 nach einem Softwareupdate über den Verdacht der Leistungsdrosselung durch Software (Softwaredrosselung) bei älteren iPhone-6-Smartphones beschwert hatten, erklärte der Hersteller Apple, dass es sich in diesem Fall um eine Schutzmaßnahme handele, da die bei älteren Akkus unvermeidbaren Kapazitätsverluste bei sonst gleicher Nutzungsintensität zum vorzeitigen Abschalten der Geräte führen könnten und liefert seit dem Bekanntwerden des Falls auch eine ausführliche Erklärung auf der Unternehmenswebseite (Apple 2022). Weil Apple seine Nutzer*innen im Vorfeld nicht ausreichend über die Softwaredrosselung informiert hat, wurde der Hersteller in manchen Ländern wegen irreführender Geschäftspraktiken und Täuschung von Konsument*innen verklagt bzw. Einigungen erzielt. Die ersten erfolgreichen Klagen gab es hierzu 2018 in Italien gegenüber Apple und Samsung, die ein ähnliche Praxis vollzogen (AGCM 2018). Eine weitere Verurteilung von Apple folgte in Frankreich (DG CCRF 2020). In den USA stimmte Apple Anfang 2020 einem Vergleich von 500 Millionen US-Dollar zu (Steinberg 2020) (siehe hierzu auch Pkt. 8.2.5).

Softwarebasiertes Batteriemangement wird von vielen Herstellern explizit als Feature beworben und soll in der Regel die mögliche Lebensdauer der Geräte und Batterien erhöhen (Pope 2019: 26). Dass es auch anders geht zeigt Apple selbst, in dem es Anfang 2020 ein Software Feature (Battery Health Management) angekündigte, das es Nutzenden mit Laptops (MacBooks) ab dem Jahr 2016 aufwärts ermöglicht, ihre Batterien länger zu nutzen, ohne dabei jedoch die Performance der Geräte zu mindern und das jederzeit deaktivierbar ist (Apple 2020).

Spätere Softwareupdates zur Performancesteuering sorgen auch in anderen Bereichen für Rechtsunsicherheit. Wie bekannt wurde, hat der Automobilhersteller Tesla Ende 2019 mit einem Softwareupdate die Batteriekapazität und damit die Reichweite einiger seiner ersten E-Autos vom Fahrzeugtyp S und X reduziert, die bis 2016 produziert wurden (Lambert 2019). Der Hersteller rechtfertigte die Maßnahme damit, dass es in einigen Modellen wohl zur Brandentwicklung kam und es sich bei der Kapazitätsbegrenzung um eine Schutzmaßnahme handelt, die zudem die Lebensdauer der Batterien verlängern soll. Aufgrund einer Klage eines Tesla Besitzers aus Kalifornien stimmte der Konzern später einem Vergleich zu, indem den betroffenen Besitzer*innen jeweils eine Entschädigung in Höhe von 625 Dollar zugesprochen wurde (Reuters 2021). Zwischenzeitlich veröffentlichte Tesla zudem ein neues Update (Software Version 2020.4.11) für aktuellere Serien der Modelle S und X, bei dem die Reichweite der Batterien wiederum erhöht werden konnte. Interessanterweise wurde hier die Effizienzsteigerung jedoch durch eine Reihe von Hardwareverbesserung in der Produktion bewirkt und musste erst durch ein nachträgliches Software-Update freigeschaltet werden (Lambert 2020). Auch in diesem Fall wird eine besondere Problematik deutlich: in der Regel werden Softwareupdates von Herstellern gemacht, um Fehler (auch „Bugs“) zu beseitigen. Im Fall von Tesla führte in dem einen Fall das Softwareupdate zur Beseitigung eines Hardwarefehlers bzw. -risikos (Entzündung älterer Batterien). In dem anderen Fall führte die Beseitigung eines Softwarefehlers bzw. die Verbesserung der Software zur Erhöhung der Reichweite.

Softwareupdates können in der Praxis beides bewirken: die Verringerung oder Verbesserung der Produktqualität und -performance. Verbraucher*innen erwarten in der Praxis demnach Vor-

und Nachteile bei Softwareupdates, die entsprechend in der Kommunikation vermittelt werden müssen. Die Beispiele von Apple und Tesla machen deutlich, dass Aussage- und Informationspflichten zum Einfluss von Software auf die Gesamtperformance hier einen Beitrag zur Rechtssicherheit auf Seiten von Hersteller und Verbraucher*innen leisten können.

5.1.4 Fall 4: Supportdauer für Softwareupdates bei Smartphones

Smartphones zeichnen sich gegenüber anderen mobilen Telefonen dadurch aus, dass sie über ein sehr flexibles anwendungsorientiertes Betriebssystem verfügen. Smartphones können heute nachträglich mit einer Vielzahl von zusätzlichen Anwendungsprogrammen („Apps“) ausgestattet werden, die es Nutzenden ermöglichen, die Geräte auch nach dem Kauf ihren Bedürfnissen anzupassen. Die Möglichkeit zur individuellen Anpassung („Customization“) ist jedoch an die Voraussetzung geknüpft, dass Nutzende ihre Apps über das Internet sicher beziehen und kompatibel mit dem jeweiligen Betriebssystem betreiben können. Zu diesem Zweck müssen mobile Betriebssysteme möglichst aktuell gehalten werden, in dem sie mit Sicherheits- und Funktionsupdates versorgt werden. In der Verfügbarkeitsdauer für Softwaresupport und bei Updates gibt es jedoch Unterschiede zwischen den verschiedenen Betriebssystemen und Herstellern (siehe Abbildung 21 rechte Spalten).

Abbildung 21: Verzug bei Sicherheitsupdates bei Betriebssystemen

Global security update availability for Smartphones (January/February 2018 Report)										
OS	Brand	Shortest time to publish a SU		Max worldwide availability delay**		SU is carrier independent for ALL devices	Support duration for security updates (2016)	Support duration for security updates (2017)		Devices SU's availability rate after 1 Month***
		For the first device	For all supported devices	Manufacturer Update	Carrier Update			Minimum	Maximum	
iOS	Apple	Day(s)	Day(s)	1 Day	-	Yes	5 years	4 years*	5 years	ALL devices
Windows	Microsoft / Nokia	Day(s)	Day(s)	1 Day	-	Yes	3 years	4 years		ALL devices
PrivatOS	Silent Circle	Weeks/Month*	N/A	1 Day	-	Yes	3 years	3 years		ALL devices
Android	Essential	Day(s)	N/A	1 Day	Month(s)*	No	N/A	3 years (Expected)*		High
	Google	Day(s)	Day(s)	2 weeks*	Month(s)	No	2 years	3 years		High
	BlackBerry	Week(s)	Week(s)	Week(s)	Month(s)	No	2 years	2 years		Medium/High
	Nokia (HMD)	Week(s)	1 Month	Week(s)	Month(s)	No	N/A	2 years (Expected)*		Medium/High
	Sony	Week(s)	Month(s)	Quarter(s)	Quarter(s)	No	1,5 years	1,5 years	2 years	Medium/High
	FairPhone	Week(s)*	N/A	1 Day*	-	Yes	1,5 years*	2 years*		ALL devices but partially updated*
	Huawei	Week(s)	Month(s)	Quarter(s)	Quarter(s)	No	1/1,5 years	1,5 years	2,5 years	Medium/Low
	LG	Week(s)	Month(s)	Quarter(s)	Quarter(s)	No	1/1,5 years	1,5 years	2,5 years	Medium/Low
	Samsung	Week(s)	Month(s)	Quarter(s)	Quarter(s)	No	1/1,5 years	1 year	2,5 years	Medium/Low
	Asus	Week(s)	Month(s)	Quarter(s)	Quarter(s)	No	1/1,5 years	1 year	1,5 years	Low
	Motorola (Lenovo)	Week(s)	Month(s)	Quarter(s)	Quarter(s)	No	1/1,5 years	1 year	2 years	Low
	OnePlus	Month(s) *	Month(s)	Quarter(s)	-	Yes	1/1,5 years	1,5 years	2 years	Low & partially updated*
	Honor (Huawei)	Month(s)	Month(s)	Quarter(s)	Quarter(s)	No	1/1,5 years	1 year	1,5 years	Low
	HTC	Month(s)	Month(s)	Quarter(s)	Quarter(s)	No	1/1,5 years	1 year	1,5 years	Low
Blu (Tinno)	Month(s)	Month(s)	Quarter(s)	Quarter(s)	No	1/1,5 years	1 year	1,5 years	None	
Wiko (Tinno)	Month(s)	Month(s)	Quarter(s)	Quarter(s)	No	1/1,5 years	1 year	1,5 years	None	

SU = Security Update. After a high or critical security breach has been unveiled.
 * Apple : They stopped supporting iPhone 5C in 2017 after 4 years, all other devices since iPhone 4S (2011) have been supported for 5 years.
 * Silent Circle announcement : "Critical vulnerabilities are patched within 72 hours of detection or reportin", but January 2018 security patch was available only after a delay of 1 month.
 * Essential : Most of US and Canadian carrier push update directly from Essential, or in only few days/weeks, but some carriers can also take months (like Telus).
 * Essential & Nokia : They started selling phones in 2017. We have indicated the official support announcement.
 * Google : Delay from official security policy <https://support.google.com/nexus/answer/4457705>
 * Fairphone : Lasts updates doesn't cover all security vulnerability for January/February (Cover only 50% high-critical security vulnerability)
 * Fairphone, duration for SU : FairPhone 1 had only 1,5 years of support (Until August 2015), FairPhone 2 had in 2017 2 years of support.
 * OnePlus: deploy partial updates for limited high-critical security updates every month. Full security update are usually every 2 months.
 ** When the first update is provided for the device by the manufacturer.
 *** Average support duration for all devices.
 Status at 15 february, more than 1 month after publication of Meltdown/Spectre security flows, and 25 criticals/High CVE.
 Author : SecX13/S.R.

Quelle: SecurityLab 2018

Hersteller wie Apple¹¹, Samsung¹² und Google¹³ veröffentlichen regelmäßig Angaben zu Support- und Updatedauern ihrer Geräte. Apple bietet mit über 5 Jahren Sicherheitsupdates für seine Geräte einen verhältnismäßig langen Service an. Demgegenüber bieten Hersteller, die auf das offene Betriebssystem Android setzen, ihren Nutzenden mit 2.5 Jahren bis 3 Jahren einen vergleichsweise kurzen Zeitraum an. Der Hersteller Fairphone, der mit seinem Fairphone 1 im Jahr 2013 eines der bis dato nachhaltigsten Smartphones vorstellte, konnte über verschiedene Modellreihen seine Software-Supportdauer sukzessive steigern. Gegenüber dem Fairphone 1 mit einem Software-Support von 3.5 Jahren erreicht der Hersteller bei der Nachfolgeneration Fairphone 2 bereits 6 Jahre und verspricht nun auch für die aktuelle Modellreihe einen Support von 5 Jahren (Wölbert 2021; Fairphone 2017). Der Hersteller Google garantiert für seine Modellreihen Pixel 3, 4 und 5 Updates für mindestens 3 Jahre. Für die aktuelle Modellreihe Pixel 6 aus dem Jahr 2021 verspricht der Hersteller 5 Jahre (Google 2022).

Das Betriebssystem Android ist eine Entwicklung von Google und wird als Open Source Variante allen Herstellern frei zur Verfügung gestellt. Möchten Hersteller jedoch die hochkompatiblen Google Dienste nutzen (Google Appstore, Maps etc.), liefert Google an die betreffenden Hersteller auch lizenzierte Versionen aus, die dann regelmäßig mit Updates von Google versorgt werden. Die Hersteller müssen sich darüber hinaus dennoch um die Integration der Updates in die Firmware ihrer Geräte kümmern. Die Firmware übernimmt dabei die Kommunikation mit der darunterliegenden Hardware-Schicht, wobei sich der Hersteller auch mit dem Lieferanten des Chip-Set besprechen muss. Dieser Prozess gestaltet sich sehr aufwändig, wird von den Herstellern sehr unterschiedlich gehandhabt und führt in der Praxis auch zur Verzögerung bei der Auslieferung von Sicherheitsupdates. Der Hersteller HTC stellt hierzu eine anschauliche Abbildung zur Verfügung¹⁴ und es gibt ein aufschlussreiches (wenn auch sehr technisches) Interview mit Greg Kroah-Hartman, dem Stable Kernel Maintainer von Android auf der Plattform Reddit¹⁵. Hartman schildert darin insbesondere die nicht immer optimalen Implementierungsansätze der Chip-Set Hersteller, die in der Praxis dann die Abstimmung zwischen Softwareprovider (Android), dem Hersteller des Smartphones und seines Lieferanten (Chip-Set) erschweren.

In der Regel verfolgen Smartphone Hersteller eine Plattformstrategie, in dem sie durch die Bündelung von Hardware-Software-Services einen „lock-in“ der Nutzer*innen forcieren (Kenney 2011). Der einfache Wechsel zwischen verschiedenen Herstellern und deren Plattformen wird dabei heute von allen größeren Herstellern durch entsprechende Migrationswerkzeuge unterstützt. Apple iOS Nutzer*innen können heute verhältnismäßig einfach mit einer Vielzahl an individuellen Einstellungen, Apps und persönlichen Daten zu Geräten mit Googles Android wechseln und andersherum. Dies zeigt die hohe technische Kompetenz und Kontrolle der Hersteller im Umgang mit den Betriebssystemen. Die Ursachen für kurze Supportdauern lassen sich demnach nicht so sehr in technischen Beschränkungen vermuten, sondern eher in den hier zu Grunde liegenden Geschäftsmodellen.

Die aktuellen Vorschläge des Bundesamts für Sicherheit und Informationstechnik (BSI) und die Vorgaben des Blauen Engels für Anwendungssoftware sehen eine Updatedauer von mindestens 5 Jahren vor (BSI 2020; Blauer Engel 2020). Eine Umfrage der Verbraucherzentrale Rheinland-Pfalz zeigt, dass dies auch dem Wunsch der Verbraucher*innen in Deutschland entspricht (vzbv

¹¹ <https://support.apple.com/en-us/HT201624>

¹² <https://security.samsungmobile.com/workScope.smsb>

¹³ <https://support.google.com/nexus/answer/4457705?hl=de>

¹⁴ <https://www.htc.com/de/go/htc-software-updates-process/>

¹⁵ https://www.reddit.com/r/Android/comments/ezrb1y/how_many_android_devices_get_os_updates_2020/

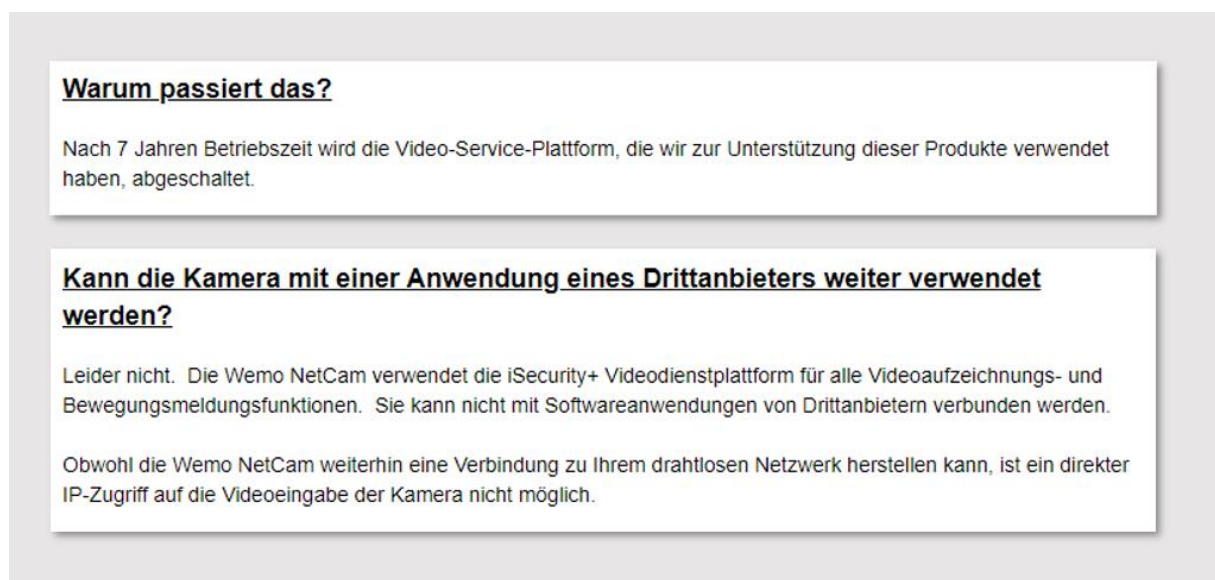
2019). Aktuelle Umfragen zeigen, dass digitale Geräte wie das Smartphone in 30-40% der Fälle aufgrund nachlassender Leistung bedingt durch abnehmende Kompatibilität zwischen aktualisierter Software und vorhandener Hardware ausgetauscht werden (vgl. VZBV, 2017; Jager-Erben & Hipp, 2017). Eine Verlängerung der Updatedauer, bei gleichzeitiger Verminderung der Performancedegradierung könnte demnach Verbraucher*innen zu längeren Nutzungsdauern ermutigen.

In einer tiefergehenden Untersuchung könnte geprüft werden, wie innerbetriebliche und marktvermittelte Hemmnisse (bspw. monopolistische Praktiken) abgebaut werden können, um längere Updatedauern zu gewährleisten oder ob womöglich die weiterführende Regulierung im Rahmen der Ökodesign-Richtlinie ausreichend Anreiz für alle Hersteller bietet, Softwareupdate und -supportdauern zu verlängern.

5.1.5 Fall 5: End of Life cloudbasierter Plattformen

Ende April 2020 informierte der Hersteller Belkin seine Nutzenden über die Einstellung der cloudbasierten Plattform iSecurity, welche zum Betrieb der Überwachungskameras der Modellreihe Wemo Netcams benötigt wird. Aufgrund fehlender Kompatibilität zu anderen Systemen oder Plattformen werden die vorhandenen Kameras damit funktionsuntauglich. Die Plattform wurde vom Hersteller ca. 7 Jahre lang betrieben. Der Hersteller selbst gibt keinerlei Gründe für die Entscheidung an und handelt zudem sehr kurzfristig, indem die genannte Plattform innerhalb nur eines Monats bis Ende Mai 2020 eingestellt wird. Der Hersteller schreibt hierzu: »[...] da die Video-Service-Plattform, die wir zur Unterstützung dieser Produkte verwenden, abgeschaltet wird.« (Belkin 2020). Betroffen von der Abschaltung der Cloud sind alle drei Modelle der Wemo NetCam Reihe: F7D7601, F7D7602, F7D7606.

Abbildung 22: Auszüge aus der End of Life Ankündigung von Belkin vom 28.04.2020



Quelle: Bildschirmkopie, <https://belkin.de>, 28.04.2020.

Vergleichbare Fälle, wie die Einstellung der Lightify-Cloud des Lampenherstellers Osram, kündigen das Support-Ende mehr als 1 Jahr im Voraus an und ermöglichen ihren Nutzer*innen zudem Alternativen durch offene Standards (Herbig 2020a). Laut Berichten von Nutzer*innen bot der Hersteller Belkin Jahre zuvor noch die Option an, seine Kameras unabhängig von der Plattform auch über eine lokale Software im Heimnetzwerk zu betreiben. Diese Möglichkeit wurde jedoch im Laufe der Zeit mit einem Firmware Update wieder genommen (Radclyffe

2021). Es stellt sich demnach die Frage, ob der Hersteller heute noch die Möglichkeit hätte per Firmware Update seinen Nutzenden den Weiterbetrieb der entsprechenden Kameras mit Software von Drittanbietern zu ermöglichen.

Im Fall Belkin ist besonders hervorzuheben, dass die Ausführung einer wichtigen Kernfunktion auf cloudbasierte Dienste ausgelagert wurde – ein Trend, der sich in Zukunft vermutlich vermehrt bei vernetzten Geräten abzeichnen kann und regulatorisch adressiert werden sollte. Hierzu zählt insbesondere ein ausreichender Abkündigungszeitraum und sofern technisch möglich, die Offenlegung von Schnittstellen zum Betrieb der Hardware mit Software von Drittanbietern sein.

5.2 Zusammenfassung und Fazit der Fallstudien

Im Bereich der Consumer-Elektronik (Haushaltselektronik, IKT, Unterhaltungselektronik) wurden für die vorliegende Studie verschiedene Fälle von softwarebedingter Obsoleszenz identifiziert und einer explorativen Analyse unterzogen. Es wurden insgesamt fünf empirische Fälle untersucht:

- ▶ Softwaregesteuerte Zählwerke bei Druckern
- ▶ Trade-up Programme bei Bluetooth-Lautsprechern unter Einsatz von softwaregesteuerten Abschaltvorrichtungen („kill switch“)
- ▶ Softwarebasierte Performance-Degradierung bei Smartphones
- ▶ Unterstützungszeitraum und Supportende bei Smartphone-Software
- ▶ End-of-Life von cloudbasierten Plattformen

Die Fallanalysen dienen in der Studie zur Validierung und Erweiterung der zu Grunde liegenden Wirkmechanismen und Entwicklung von praxisorientierten Handlungsempfehlungen.

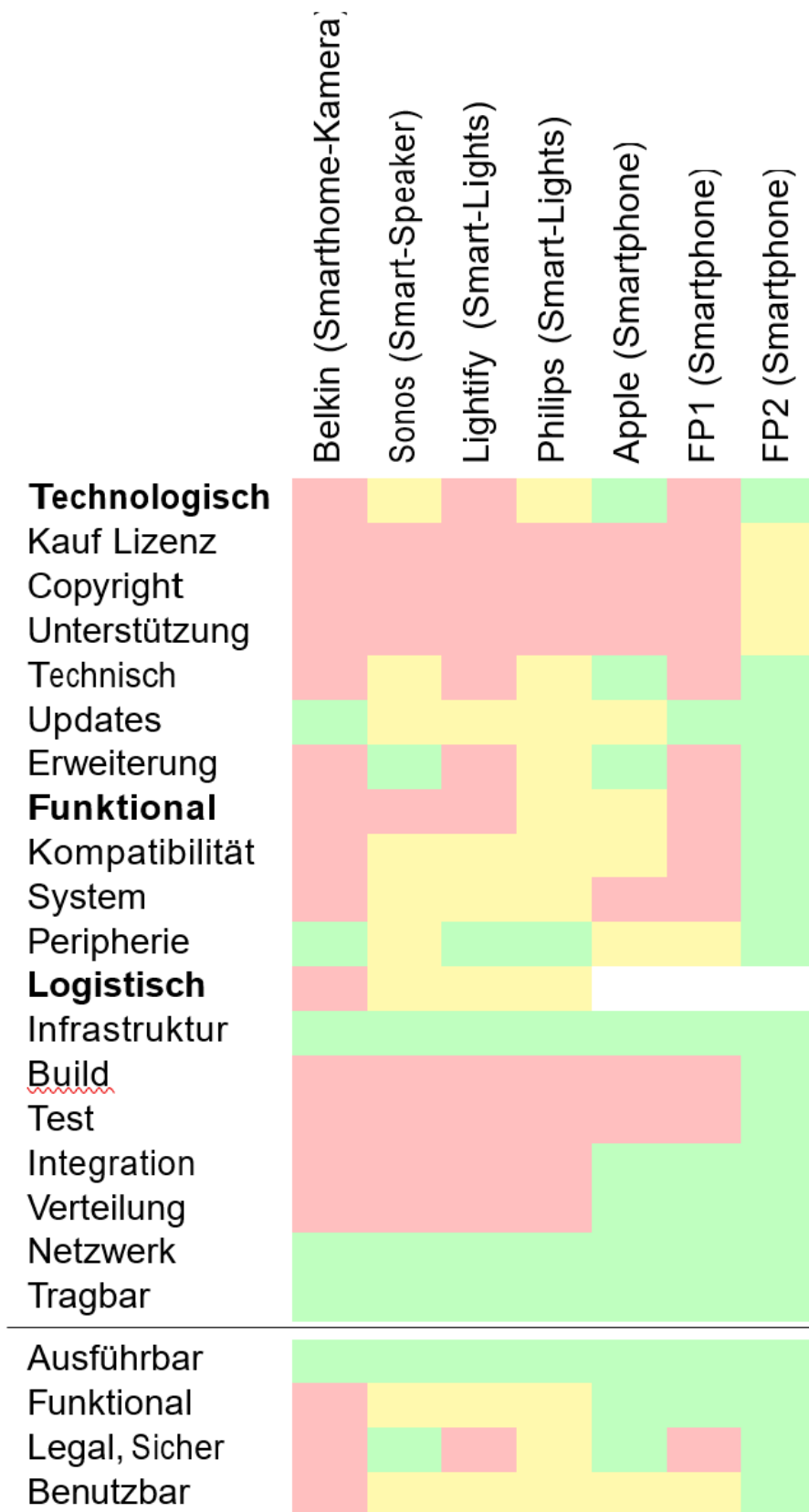
In der Einzelfallbetrachtung fällt auf, dass Hersteller softwarebasierte Funktionsbeschränkungen und Konfigurationen überwiegend mit dem Hinweis auf qualitätssichernde Maßnahmen und Produktsicherheit rechtfertigen. Hierzu zählen exemplarisch Zählwerke und Verbrauchsmesser für Druckerpatronen, die ein optimales Druckerergebnis gewährleisten sollen; die softwarebasierte Serialisierung von Einzelkomponenten, um den Einbau von Ersatzteilen von Drittanbietern zu unterbinden und die softwarebasierte Performance-Drosselung zur Verlängerung der Akkulaufzeit. In einem Fall wurde eine softwaregesteuerte Abschaltvorrichtung („kill switch“) dafür eingesetzt, mit Zustimmung der jeweiligen Kund*innen gebrauchte Produkte vollständig funktionsunfähig zu machen, um hierfür einen Gutschein für den Kauf eines Neuprodukts („Trade-In“) zu erhalten. Es lässt sich feststellen, dass die angemessene Anwendung von Schutzvorrichtungen durch Software in Geräten bisher eine rechtliche Grauzone darstellt und überwiegend im Ermessen des Herstellers liegt. In einigen Fällen kann dies jedoch den vorzeitigen Funktionsverlust von Hardware bewirken. Es gilt daher vertiefend zu klären, wie viel Schutz vor Übernutzung und Falschnutzung eine funktionsbegrenzende Intervention von Software aus Sicht des Verbraucher- und Umweltschutzes überhaupt rechtfertigt.

In der vergleichenden Fallanalyse konnte weiter festgestellt werden, dass Hersteller bisher nur unzureichend über Softwareänderungen informieren, die einen Einfluss auf die Funktionalität und Performance des Gerätes haben können – eine Praxis, die in einigen Ländern bereits zu erfolgreichen Klagen gegen Herstellende wegen Verbraucher*innentäuschung geführt hat. Hersteller sind bisher relativ frei, was die Ausgestaltung der Supportzeiträume für Software angeht. Eine vergleichende Betrachtung des Smartphone Marktes zeigt, dass die Zeiträume für den Softwaresupport generell sehr unterschiedlich sind. In einem spezifischen Fall konnte gezeigt werden, dass ein Hersteller die Unterstützung für eine Cloud-Plattform innerhalb eines

Monats nach Ankündigung eingestellt hat, ohne den Nutzer*innen eine hinreichende Begründung oder Alternativen für die mit dem Betrieb verbundenen Geräte anzubieten.

In der vertiefenden systematischen Analyse zeigt die Darstellung der Fälle als Heatmap die aufsteigende Hierarchie der Dimensionen der Softwarequalität (Abbildung 23). Der Einfluss der Ursachen auf die Funktionalität oder sogar die Ausführbarkeit tritt seltener ein, Sicherheit oder Benutzbarkeit hingegen sind häufig beeinträchtigt.

Abbildung 23: Heatmap der zusammengefassten Ursachen (oberer Teil) und Wirkungen (unterer Teil) der untersuchten Fallstudien



Quelle: Hahn 2020.

Der Vergleich der Fallstudien zeigt, dass eine Abnahme der Softwarequalität über einen längeren Zeitraum in den meisten Fällen vom Hersteller nicht verhindert, jedoch verantwortlich begleitet und z.B. durch frühzeitige Kommunikation vorgebeugt werden kann. Es bleibt jedoch im Rahmen der Definition und damit subjektiven Empfindens offen, wann dies als softwarebedingte Obsoleszenz eingestuft werden kann. Dies wird am Beispiel der Leistungsreduktion bei Smartphones deutlich: während für einige Nutzende eine Verlangsamung des Geräts ein ausreichender Grund zum Austausch ist, nehmen andere diese Leistungsreduktion in Kauf.

Zusammenfassend zeigt die Fallanalyse, dass trotz des wachsenden Einflusses von Software bei elektronischen Produkten im Rahmen der Untersuchung nur wenige stichhaltige Fälle ermittelt werden, bei denen software-relevante Aspekte die Wahrscheinlichkeit für die Obsoleszenz des Produkts erhöhen. Ein Befund, der ebenfalls von den Ergebnissen der repräsentativen Befragung im Rahmen der Studie gestützt wird (siehe Kapitel 6.1.2). Die Fallstudien belegen jedoch auch, dass die Ausgestaltung und Zusicherung von funktionsrelevanten Softwareeigenschaften bisher schlecht reguliert, weitestgehend den Herstellern überlassen sind und hieraus eine erhebliche Informationsasymmetrie gegenüber Nutzer*innen entsteht, die bisher wenig darüber informiert werden, welche Auswirkungen Softwareänderungen auf die Geräteperformance haben und wie lange sie mit einer hochwertigen Softwareunterstützung und der Nutzung der Cloud-Dienstleistung rechnen können.

6 Wahrnehmungen und Erfahrungen von Konsument*innen

Die vorangehenden Kapitel 4 und 5 haben zweierlei deutlich gemacht: Zum einen steigt der Anteil smarter Geräte in Haushalten und immer mehr Menschen sehen in der zentralen und digitalen Steuerung technologischer Geräte und Systeme im eigenen Haushalt einen Vorteil. Zum anderen finden sich vermehrt Fälle, in denen die Software einen Nutzungsdauerverkürzenden Effekt hat und Kund*innen und Konsument*innen frustriert feststellen müssen, dass ein technisch einwandfreies Gerät nicht mehr funktioniert. Doch wie häufig erleben Nutzer*innen smarter Geräte Phänomene von Softwareobsoleszenz tatsächlich und wie gehen sie damit um? Aktuelle Umfragen zeigen, dass digitale Geräte wie das Smartphone in 30 bis 40% der Fälle aufgrund nachlassender Leistung bedingt durch abnehmende Kompatibilität zwischen aktualisierter Software und vorhandener Hardware ausgetauscht werden (vgl. VZBV, 2017; Jäger-Erben & Hipp, 2017). Würde sich dieser seit Jahren existierende Trend bei IKT auch auf vernetzte Haushaltsgeräte übertragen, so werden auch bisher eher langlebige Geräte immer kürzer genutzt werden können.

Bisher liegen keine systematischen Umfragen zur Nutzer*innen-Perspektive auf softwarebedingte Obsoleszenz vor. Diese Forschungslücke sollte das Projekt SobO beginnen zu schließen. Die Herausforderungen für das Design der Umfrage bestanden darin, dass bisher eine eher geringe Sensibilisierung oder gar Unkenntnis gegenüber der Problematik softwarebedingter Obsoleszenz zu erwarten ist. Zwar spielt das Thema in den Medien immer wieder eine Rolle, es ist jedoch anzunehmen, dass die eigene Wahrnehmung eher von den öffentlichen Debatten geprägt ist, als dass eigene Erfahrungen mehr oder weniger objektiv berichtet werden. Darüber hinaus sind Erfahrungen im Kontext alltäglicher Routinen der Gerätenutzung generell schwer zu erfragen, die vergleichsweise eher geringe Qualität der Daten in Online-Umfragen kann zudem die Interpretierbarkeit der Daten erschweren. Daher waren eine planvolle und systematische Vorbereitung und Entwicklung des Fragebogens wichtig, die im Folgenden zunächst beschrieben wird.

6.1 Fragestellungen und Vorgehen

Die Basis für die Entwicklung des Fragebogens bildete auf der einen Seite eine umfassende Recherche bisheriger Forschungsarbeiten sowie relevanter statistischer Daten zur Ausstattung und den Nutzungserfahrungen mit vernetzten Geräten. Außerdem wurde ermittelt, welche vernetzten Geräte im Bereich Consumer Electronics im Handel derzeit verfügbar sind. Ein wichtiger Bestandteil der Recherche war zudem, in Nutzer*innen-Foren und auf Produktbewertungsseiten nach berichteten Problemen zu suchen und diese zu systematisieren. Folgende Fragen standen im Vordergrund:

- ▶ Was sind Ergebnisse empirischer Forschung bzw. statistischer Erhebungen zum Konsum vernetzter Geräte? Z.B. genutzte Gerätetypen, Haushaltsausstattung, Kaufverhalten, Motive der Nutzung, Nutzungserfahrungen, etc.
- ▶ Welche Probleme berichten Nutzer*innen im Zusammenhang mit der Software von Geräten? Wie beeinträchtigen softwarebedingte Probleme die Nutzung

Tabelle 14 zeigt die Schwerpunkte der Vor-Recherche.

Tabelle 14: Schwerpunkte der Vor-Recherche zur Entwicklung des Fragebogens

Quellen	Erkenntnisinteresse
Studien und Statistiken (z.B. statista, destatis)	Ausstattung mit vernetzten Geräten, Verkaufszahlen (ggf. unter Berücksichtigung sozio-ökonomischer Aspekte)
Fachhandel bzw. Online-Elektronik-Handel (z.B. Online-Angebot von Saturn)	Angebot von vernetzten Geräten: Welche Geräte gibt es überhaupt im Handel?
Fachartikel, Berichte aus empirischer Forschung (google scholar)	Existenz von Forschung sowie Forschungsstand zum Thema Software-Nutzung und Nutzer*innen-Erfahrungen im Kontext softwarebedingter Obsoleszenz;
Nutzerforen im Internet (z.B. chip.de)	Problemauffassung und -beschreibung, Ermittlung von Praxisbeispielen, Schwerpunktthemen und -problembereiche
Konsumenten-Reviews von vernetzten Geräten (z.B. amazon)	Qualitativer Einblick: Welche Probleme werden geschildert? Welcher Kenntnisstand wird deutlich? Welche Erwartungen treffen auf welche Herausforderungen?

Die Recherchen bildeten den Hintergrund für die Fragestellungen und Frageformate im Fragebogen. Die Betrachtung des Handelsangebots sowie der bisherigen Erhebungen zur Geräteausstattung stellten die Basis für Abfrage der besessenen Geräte

Die Recherche nach Studien und Statistiken ergab, dass es vorwiegend deskriptive Erhebungen zur Geräteausstattung von Haushalten, zu den Erwartungen und dem soziodemographischen Hintergrund der Nutzer*innen vernetzter Geräte sowie zu Ausstattungswünschen gibt. Aktuell finden sich besonders zahlreiche Erhebungen zum Stichwort Smart Home, wobei dieser Begriff sehr unterschiedlich definiert wird. In einigen Untersuchungen zählen beispielsweise vor allem Produkte und Dienstleistungen der Hausautomatisierung dazu (Statista 2019), in anderen Erhebungen werden auch per App-steuerbare Stand-Alone-Geräte wie Smart Speaker und SmartTvs hinzugezählt (Deloitte 2019). Die Mehrzahl der Studien bzw. Erhebungen sind dem Bereich der Marktforschung zuzuordnen, eine intensivere wissenschaftliche oder analytische Auseinandersetzung mit den Nutzer*innen-Erfahrungen im Kontext softwarebedingter Obsoleszenz scheint nicht stattzufinden. Die neuesten Studien finden sich auf dem Service-Portal Statista (z.B. Statista 2018, Breitkopf 2019), werden von Beratungsunternehmen wie Deloitte (2018) umgesetzt oder vom Statistischen Bundesamt zur Verfügung gestellt (Statistisches Bundesamt 2019). Auch bezüglich der Daten ergibt sich kein einheitliches Bild. So unterscheiden sich die teilweise die Ausstattungsraten mit spezifischen Smart-Home-Anwendungen erheblich (z.B. 10% smarte Beleuchtung bei Kunst, 2019 im Vergleich zu 34% bei Breitkopf, 2019) und auch zu den Nutzer*innen-Profilen finden sich widersprüchliche Ergebnisse. Eine Untersuchung von Statista (2019) sieht vor allem Haushalte mit niedrigen Einkommen als Nutzer*innen von Smart-Home-Anwendungen während bei Deloitte (2018) vor allem Menschen mit hohem Einkommen solche Produkte und Dienstleistungen nutzen.

Neben der definitorischen Unschärfe der verwendeten Begriffe und Kategorien scheint auch der Markt sehr dynamisch zu sein, so dass sich ein eher heterogenes Bild der Konsument*innen-Markts ergibt. Vor diesem Hintergrund tragen die Ergebnisse des Projekts erheblich zum bisher eher dürftigen Stand des Wissens bei.

6.1.1 Qualitative Untersuchung Online-Foren und Produktbewertungen

Eine weitere wichtige Quelle war die qualitative Auswertung von Nutzer*innen-Beiträgen in Online-Foren oder auf Bewertungsplattformen. Hier wurde auf Basis einer Reihe von Keywords

(zunächst verschiedene Gerätebezeichnungen, im weiteren Verlauf auch Problemkategorien wie „keine Verbindung mehr“, „keine Updates mehr“, „Fehlermeldungen“ oder „Software-Problem“) nach relevanten Beiträgen gesucht. Die folgenden zwei Abbildungen zeigen Beispiele für relevante Erfahrungen von Nutzenden.

Abbildung 24: Erfahrungen von Nutzenden aus Recherche in Foren 1

Nutzer*in A auf Bewertungsforum 1 am 21. Januar 2020

„Sonos Empfehlung 5 Jahre alte Geräte für den Recyclinghof“ (1 von 5 Sternen)

Insgesamt besitzt ich zwei Sonos Play 5 erste Generation, sowie ein Sonos connect, und einen neueren Sonos Play 1. Bisher war ich mit dem System ganz zufrieden, auch es es auf verständliche Art kein Bluetooth hatte. Dies hat sich jedoch seit dem 20. Januar 2020 grundlegend geändert. Sonos hat mir in einer Mail mitgeteilt, dass meine play 5 sowie mein Sonos connect zukünftig nicht mehr mit Updates unterstützt werden. Jetzt kommen die ersten schlaun Kritiker und weisen mich darauf hin, dass das System auch zukünftig weiter benutzt werden kann. Stimmt, nur Sonos spricht von „erheblichen“ Einschränkungen“. Dies könnte zum Beispiel sein, dass die Schnittstelle Spotify neu definiert wird zukünftig nicht mehr auf dem alten System läuft. Aber einen kurzen Abriss zur Geschäftspolitik von Sonos:

Herbst 2019: Musik, die auf einem Apple Gerät gespeichert ist, kann nicht mehr über Sonos abgespielt werden. Das betrifft auch gekaufte Titel bei iTunes.

IOS 10.3.3 wird nicht mehr unterstützt. Wohl gemerkt, diese Version war, nach meinem Kenntnisstand bis 22. Juli 2019 aktuell. Hier ist Sonos wirklich rasend schnell. Laut Sonos gibt es erhebliche Einschränkungen wie zum Beispiel, es können mit den Lautsprechern keine Gruppen mehr gebildet werden, die Musik-Bibliothek kann nicht mehr verwaltet werden, keine neuen Lautsprecher hinzugefügt werden. Das war es dann wohl für mein iPad 4.

Abbildung 25: Erfahrungen von Nutzenden aus Recherche in Foren 2

Nutzer*in B auf Frageforum 2 am 1. Januar 2013

Hallo, guten Tag und eine frohes neues Jahr nachträglich.

Ich kann auf meiner Nintendo Wii kein Update durchführen. Das ist dahingehend auch ärgerlich, weil ich ohne ein Update nicht mehr auf den Shop komme.

Das Update läuft einige wenige Minuten. Der Fortschrittsbalken bewegt sich dabei kaum. Dann kommt die Meldung mit dem Fehlercode 32002 und er Meldung, dass doch mal die Internetverbindung überprüft werden soll. Kurz vor dem Updateversuch wird aber die Verbindung zum Router (FritzBox 7390) und zum Internet getestet und nach wenigen Sekunden kommt dort die Meldung, dass alles ok ist. Aber das Update selber, läuft einfach nicht durch.

Die Verbindung zwischen Wii und Router läuft per Wireless. Der Router ist zwischen 2-3 Meter von der Wii entfernt. Die Wii selber hat vier oder fünf Jahre auf dem Buckel und ich habe auch schon erfolgreich Updates durchgeführt. Das letzte Mail ist aber auch schon lange her, mindestens ein Jahr. Bei einer anderen Hardware hatte ich auch schon Probleme, mit meinem Firmware-Update, da die vorher installierte Firmware-Version zu alten gewesen ist. Könnte das auch hier der Fall sein?

Die Forenbeiträge wurden zum einen pro Geräte und Gerätekategorie ausgewertet, zum anderen wurde versucht, übergreifende Dimensionen und Kategorien im Hinblick auf die Fragestellungen (Was wird erlebt? Wie wird damit umgegangen?) zu identifizieren. Die Ergebnisse flossen insbesondere in die Formulierung von Fragen und Antwortkategorien des Fragebogens ein.

6.1.2 Repräsentative Konsument*innen-Befragung

Im Rahmen des Projekts wurde eine Online-Umfrage im Umfang von 30 Minuten mit einer Stichprobe von 1.000 Teilnehmenden im Alter von 18 bis 75 durchgeführt. Im Mittelpunkt standen der Geräte-Besitz und die Nutzungspraktiken bei vernetzten Geräten, die Umgangsweisen und Erfahrungen mit softwarebedingten Problemen bzw. Einschränkungen in der Funktionalität und Geräte-Performance. Außerdem sollten relevante Konsum-Bedürfnisse und Erwartungen zur Konsument*innen-Souveränität im Kontext von vernetzten Geräten erhoben werden.

Die erhobenen Daten wurden vorwiegend deskriptiv ausgewertet. Der verwendete Fragebogen befindet sich im Anhang C.

6.2 Ergebnisse der Auswertung von Online-Foren und Produktbewertungen

Insgesamt wurden ca. 50 Fälle von Problembeschreibungen ausgewertet. Am häufigsten werden Verbindungsprobleme berichtet. So lassen sich eine Vielzahl von den beschriebenen Geräten zum Teil bereits nach kurzer Zeit nicht mehr mit dem Netz, mit anderen Geräten oder mit der Steuerung verbinden. Die Verbindungsprobleme mit anderen Geräten werden zum Teil auf Inkompatibilitäten mit anderen (älteren Versionen von) Betriebssystemen zurückgeführt. Weitere Probleme sind, dass keine Updates mehr durchgeführt werden, dass das Gerät unverständliche oder unerklärliche Fehlermeldungen zeigt oder auch plötzliches Ausschalten häufig vorkommt. Zu beobachten war darüber hinaus eine hohe Emotionalität der Beiträge. Das heißt, der Ärger über ein nicht funktionierendes Gerät ist stark ausgeprägt. Die Auswertung der Foren-Beiträge sollte vor allem die Entwicklung von Items des Fragebogens unterstützen. Die hier aufgeführten Beobachtungen stellen nur Ausschnitte dar, die vermutlich eher Einzelfälle sind.

Tabelle 15: Auswertung Nutzer*innen-Erfahrungen aus Online-Foren und Produktbewertungen

Gerät	Kritik	Einordnung
Belkin Netcam (Überwachungskamera mit Cloudanbindung)	Geräte funktionieren nur mit Cloudanbindung, welche vom Hersteller nach 7 Jahren abgeschaltet wurde. Verbraucher*innen sind über die Obsoleszenz der Geräte verärgert und fordern politische Maßnahmen für die Zukunft. Für einen Großteil der genutzten Funktionen sei die Cloud nicht prinzipiell notwendig, Unverständnis für ausbleibende Maßnahmen des Herstellers, zumindest Grundfunktionen für die Zeit nach Cloudabschaltung zu ermöglichen.	Markanter Fall von softwarebedingter Obsoleszenz, da die Abschaltung der Cloud zum kompletten Ausfall der Geräte führte und nur nicht zu einer schleichenden Verschlechterung der empfundenen Nutzungsqualität.
Sonos Smart Speaker (Netzwerk-Lautsprecher für Haushalte, die über WLAN bedient werden)	Softwareänderungen führen zu Inkompatibilitäten zwischen unterschiedlichen Produktreihen. Ältere Lautsprecher sind mit neuen Softwareversionen auf Smartphones und anderen Sonos-Geräten nicht mehr nutzbar.	Schnittstellenänderungen erfordern tendenziell leistungsfähigere Hardware. Ältere Geräte sind dann nicht mehr kompatibel. Technische und produktpolitische Gründe für die Änderungen sind von außen nicht unterscheidbar.

Gerät	Kritik	Einordnung
Sonos (Fortsetzung)	Verbraucher*innen hatten längere Nutzungszeiträume erwartet – insbesondere mit Verweis auf die hohen Anschaffungskosten - und kritisieren die Produktpolitik des Herstellers.	Smart-Home-Systeme werden mit Netzwerkfunktionen beworben (z.B. Lautsprecher in mehreren Räumen, die dieselbe Musik synchron abspielen). Softwareupdates können zu Inkompatibilität vorher kompatibler Geräte im Netzwerk führen.
Osram Lightify Gateway (Basisstation zur Steuerung von Smart-Home-Beleuchtung)	Das Gerät unterstützt nur ältere WLAN-Standards mit beschränkter Bandbreite und nicht alle Funkkanäle.	WLAN-Standards sind komplex und werden stetig weiterentwickelt. Geräte sind abwärtskompatibel, für manche Funktionen. Für hohe Bandbreiten werden aber neuere Geräte oder Standards benötigt. Ältere Geräte können unter Umständen alle Geräte im Netzwerk auf den gemeinsamen, leistungsschwachen Standard zwingen.
Osram (Fortsetzung)	Abschaltung der Cloud-Server für 2021 angekündigt, dadurch wird die Funktionalität eingeschränkt.	Cloud-Infrastruktur ist außerhalb der Kontrolle der Nutzer*innen
Philips Hue (Produktreihe für Smart-Home-Beleuchtung)	Ältere Geräte werden nicht mehr unterstützt, dies wird als Lock-In-Effekt und impliziter Zwang zum Kauf eines Ersatzgeräts interpretiert.	Technische und produktpolitische Gründe für die Änderungen sind von außen nicht unterscheidbar. Klare Informationen zur voraussichtlichen Lebensdauer zum Kaufzeitpunkt fehlen
Philips (Fortsetzung)	Anzahl der ansteuerbaren Geräte im Netzwerk zu klein für größere Installationen, nach Herstellerangaben wegen des begrenzten Speicherplatzes der Basisstation	Smart-Home-Geräte verfügen häufig über leistungsschwacher Hardware. Technisch nicht versierte Nutzer*innen sind bei umfangreichen Installationen überfordert.
Philips (Fortsetzung)	Unerwartete Inkompatibilitäten zwischen Geräten der Hue-Reihe	Bei großen Produktlinien mit mehreren Hardwaregenerationen ist eine Kompatibilität oft nicht gegeben. Nutzer*innen erwarten adressatengerechte und leicht auffindbare Informationen
Philips (Fortsetzung)	Standortdatenerhebung und Erstellung eines Kundenkontos als Voraussetzung für den Betrieb	Nutzer*innen verlangen nach Datenschutz und Erklärungen, wofür erhobene Daten benötigt werden
Apple Iphone 7 (Smartphone)	Telefon nach Softwareupdate mit großem Energieverbrauch auch ohne intensive Benutzung und einhergehend mit eingeschränkter Benutzbarkeit.	Durch Updates können die Anforderungen an die Leistungsfähigkeit der Hardware steigen. Dadurch steigt auch der Energieverbrauch..

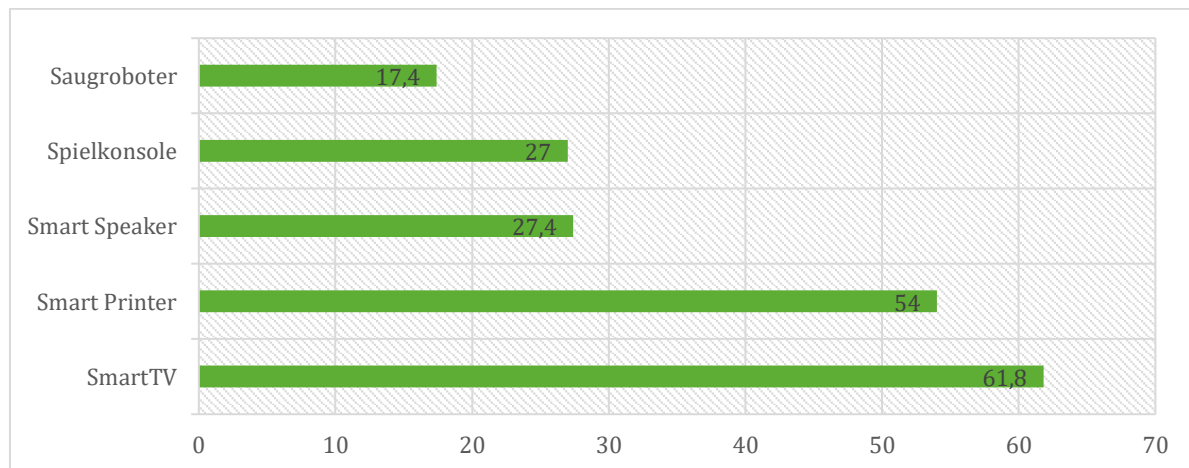
Gerät	Kritik	Einordnung
Fairphone 1 (Smartphone)	Entgegen den Versprechen sind nicht alle Ersatzteile über viele Jahre lieferbar: Kund*innen berichten von Problemen durch unzureichende Befestigung im Gehäuse des angebotenen Ersatzakkus mit anderem Einbaumaß	Besonders große Enttäuschung, da die versprochene Ersatzteilversorgung ein Hauptgrund für den Kauf des Telefons war.
Fairphone 1 (Fortsetzung)	Kundendienst des Herstellers nur in englischer Sprache erschwert den Kontakt.	Nutzer*innen erwarten adressatengerechte und leicht auffindbare Informationen

6.3 Ergebnisse der repräsentativen Befragung

Die Umfrage wurde im März 2020 umgesetzt. Es nahmen 1.000 Personen im Alter von 18-75 teil, das mittlere Alter war 46,9 Jahre, die Hälfte der Befragten war weiblich. Ca. ein Drittel der Befragten hatte einen Hauptschulabschluss, ein Drittel die Mittlere Reife, die restlichen Befragten mindestens das Abitur. Die Einkommensverteilung entsprach der Verteilung in der Gesamtbevölkerung.

Bei der Frage nach der aktuellen Gerätenutzung zeigte sich, dass Smartphones und Internet-Router in der Nutzung am meisten verbreitet und in ca. 90% aller Haushalte vorhanden sind. Gefolgt werden diese von Notebooks und Tablets, die in über 60% aller befragter Haushalte vorhanden sind. Die weiteren Geräte mit nennenswerten Häufigkeiten werden in Abbildung 26 aufgezeigt. Smarte Beleuchtungen, Steckdosen, Heizungen und Sicherheitssysteme werden in 10 bis 20% der Fälle genutzt. Eher selten finden sich digitale Kühlschränke und Waschmaschinen (unter 5%).

Abbildung 26: Smarte Geräte im Haushalt [%] (ausgenommen Smartphone, Notebook, Tablet)



Quelle: Eigene Darstellung

Zu den am häufigsten vorhandenen Gerätekategorien wurden die Nutzungsdauern abgefragt: Internet-Router werden typischerweise vier Jahre und länger genutzt, während Smartphones typischerweise bis zu drei Jahre genutzt werden. Fast doppelt so viele der Befragten (316) gaben an, ihren Internet-Router 4 und mehr Jahre zu nutzen als 3 Jahre und weniger (144).

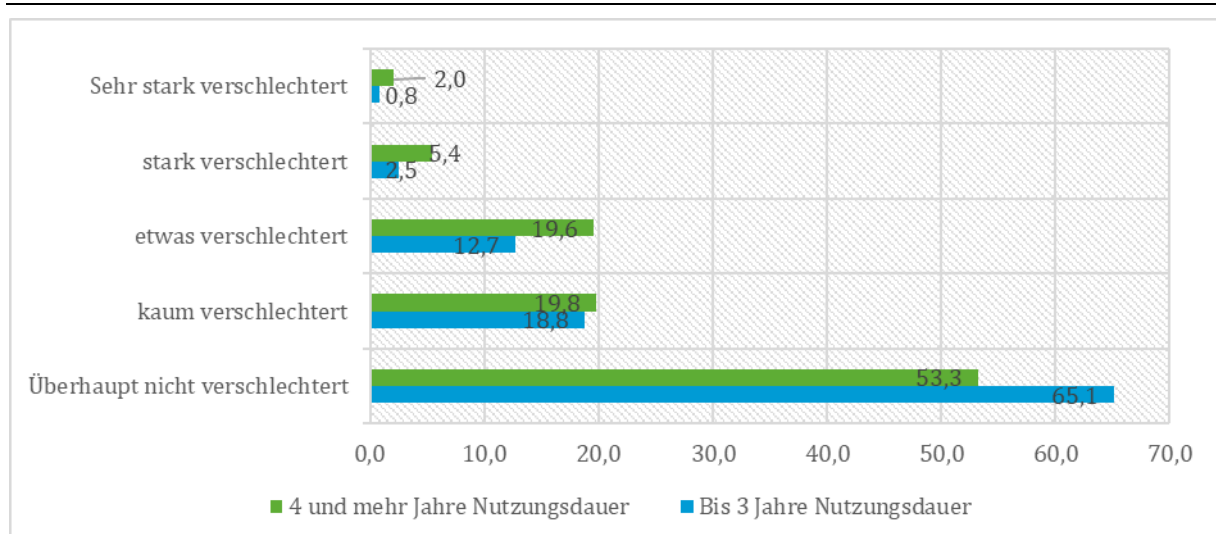
6.3.1 Erfahrungen und Umgang mit Problemen

Bei einer generellen Betrachtung zeigt sich, dass die Befragten in der Regel eher wenige Probleme mit ihren Geräten haben. Bei fast allen Geräten geben die meisten Befragten an, dass die Leistung oder Bedienbarkeit sich überhaupt nicht oder kaum verschlechtert habe. Den höchsten Mittelwert erreichen digitale Kühlschränke mit 2,17 (auf einer Skala von 1 = überhaupt nicht verschlechtert bis 5 = sehr stark verschlechtert), dies aber bei sehr geringen Fallzahlen (unter 30 Fälle). Ansonsten werden vor allem bei Smartphones, Internet-Routern, Notebooks und Tablet Verschlechterungen berichtet (wie langsamer werden, erhöhter Energieverbrauch). Falls technische Einschränkungen festgestellt werden, nutzen die meisten Befragten die Geräte trotzdem weiter und tolerieren die Einschränkungen. Die Geräte werden nur in den wenigsten Fällen entsorgt oder ersetzt. Auch eine Reparatur (professionell oder selbst durchgeführt) kommt für die meisten Befragten nicht in Frage. Die meisten Befragten beschwerten sich auch nicht auf Kundenportalen. Dies betrifft aber die zusammengefassten Werte unabhängig von der bisherigen Nutzungsdauer. Es kann vermutet werden, dass Probleme erst auftreten, wenn beispielsweise Gewährleistung und Herstellergarantie nicht mehr in Anspruch genommen werden können.

Im Folgenden werden die Daten daher für verschiedene Gerätekategorien noch einmal gezielt im Verhältnis zur Nutzungsdauer betrachtet. Außerdem werden Unterschiede in der Problemwahrnehmung nach Lebensalter der Befragten betrachtet.

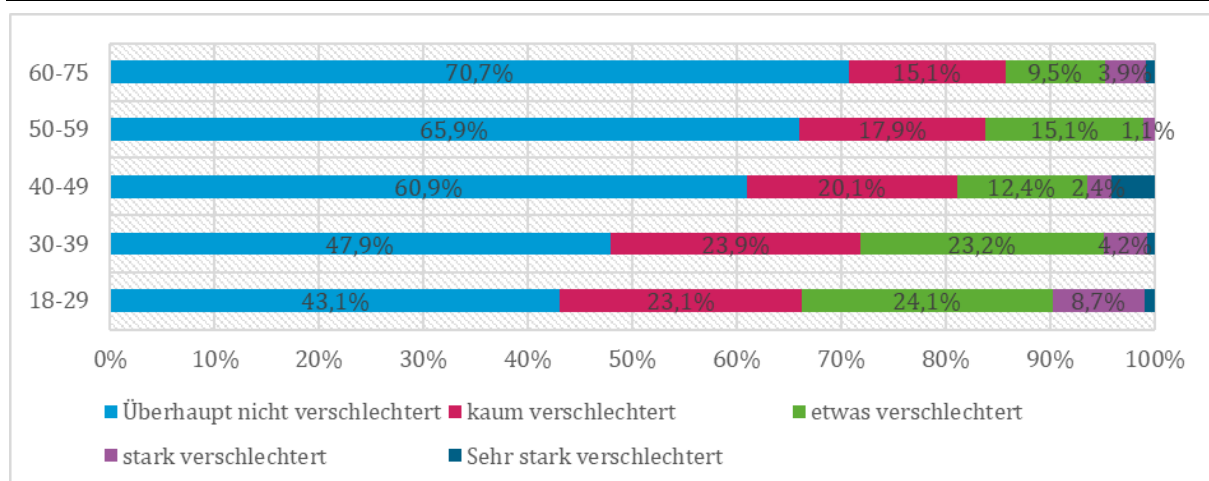
Bei **Smartphones** nehmen die wahrgenommenen Verschlechterungen mit zunehmender Nutzungsdauer leicht zu. Die Nutzungsdauer hat dabei fast keinen Einfluss auf den Umgang der Nutzer*innen mit Problemen oder das In-Betracht-ziehen einer Eigenreparatur. Bei der Gegenüberstellung der Altersgruppen ist zu sehen, dass die **höheren Altersgruppen zunehmend überhaupt keine Verschlechterung** des Gerätes feststellen. Möglicherweise sind bei jüngeren Altersgruppen die Erwartungen an das Gerät oder aber auch die Nutzungsintensitäten größer.

Abbildung 27: Wahrgenommene Verschlechterungen beim Smartphone im Verhältnis zur Nutzungsdauer [%]



Quelle: Eigene Darstellung

Abbildung 28: Wahrgenommene Verschlechterungen beim Smartphone im Verhältnis zur Altersgruppe der Nutzer*innen



Quelle: Eigene Darstellung

Bei einer Gegenüberstellung der beiden Gruppen (Nutzungsdauer bis zu 3 Jahre vs. Nutzungsdauer 4 und mehr Jahre) bezüglich möglicher Arten mit Problemen umzugehen, zeigen sich **leichte Verteilungsunterschiede**: Nutzer*innen von eher jüngeren Geräten holen sich beispielsweise eher Hilfe aus dem Bekannten- und Familienkreis oder informieren sich im Internet über Möglichkeiten, mit dem Problemen umzugehen.

Beim Internet-Router gaben die meisten Personen (538 von 879) an, dass sie überhaupt keine Verschlechterung wahrgenommen hätten. Aber auch hier zeigt sich wieder die stärkere Problemwahrnehmung in den jüngeren Generationen. Während die Angabe, überhaupt keine Verschlechterungen beim Gerät festzustellen, mit dem Alter zunimmt, werden bei zunehmend jüngeren Befragten häufiger „etwas Verschlechterungen“ und „starke Verschlechterungen“ wahrgenommen.

Beim **Notebook** geben jeweils prozentual mehr Befragte an, Verschlechterungen wahrzunehmen, wenn die Nutzungsdauer länger als 4 Jahre ist. Ergänzend geben prozentual mehr „Kurzzeit-Nutzer*innen“ an überhaupt keine Verschlechterungen wahrzunehmen.

Beim Umgang mit den Verschlechterungen unterscheiden sich die Nutzer*innen nach Nutzungsdauer. So stimmen Nutzer*innen jüngerer Geräte stärker zu, Geräte trotz Einschränkungen zu nutzen, wohingegen Nutzer*innen älterer Geräte sich eher Hilfe aus dem Bekannten- und Familienkreis holen.

Im Vergleich der Altersgruppen bei der Notebooknutzung fällt wie weiter oben beschrieben auch hier auf, dass mit zunehmendem Lebensalter der Befragten eine Verschlechterung des Gerätes seltener wahrgenommen wird.

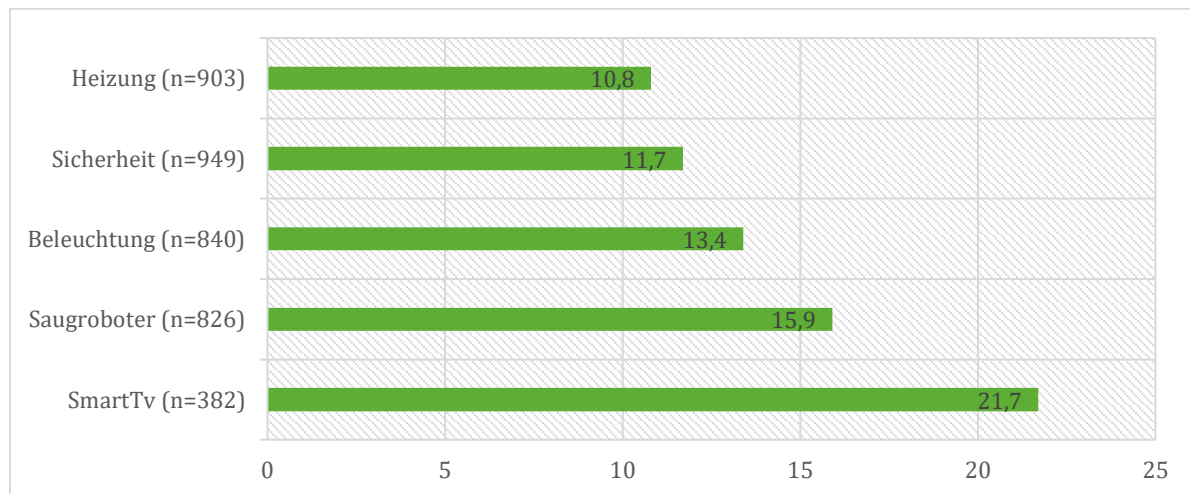
Auch beim **Smart Speaker** treten wahrgenommene Verschlechterungen eher bei längerer Nutzungsdauer ein. Allerdings sind die Fallzahlen für diese Gerätekategorie gering, wodurch die Aussagekraft eingeschränkt ist. Wiederum nehmen Jüngere eher Verschlechterungen wahr als Ältere.

6.3.2 Konsumwünsche

Die Frage nach den vermutlichen Anschaffungen in der kommenden Zeit kann Aufschluss über Markttrends und die Entwicklung der Haushaltsausstattung geben. Die Mehrheit der Befragten besitzt bereits einen Smart-TV, dennoch steht diese Gerätekategorie an der Spitze der

wahrscheinlichen Anschaffungen - ca. ein Fünftel bewertet eine baldige Anschaffung als wahrscheinlich. Zunehmen dürfte zudem der Besitz von Saugrobotern. Bei den Smart-Home-Anwendungen ist das Interesse vor allem auf Beleuchtung, Sicherheit und Heizung gerichtet.

Abbildung 29: Die am häufigsten genannten Anschaffungswünsche, Angaben in % (das n in Klammern gibt an, wie viele Befragte das Gerät noch nicht besitzen und daher die Frage gestellt bekamen)



Quelle: Eigene Darstellung

6.3.3 Kompetenz beim Einkauf und Nutzung vernetzter Geräte

Um zu ermitteln, welche Informationen und Rechte sich Nutzer*innen wünschen, wurde die Zustimmung zu verschiedenen Kategorien der Konsument*innen-Souveränität abgefragt. Am stärksten werden mehr Informationen zur Haltbarkeit und zur Software-Unterstützung von Geräten gewünscht (ca. zwei Drittel der Befragten). Mehr Wissen zur sachgemäßen Pflege und Wartung, mehr Unterstützung bei Fehlern und Einschränkungen und mehr Informationen durch den Handel und die Hersteller wünscht sich über die Hälfte. Mehr Rechte wünschen sich etwas weniger als die Hälfte, das ist auch bei den Informationen über die Umweltbelastungen so. Knapp 60% wollen hingegen mehr darüber wissen, was nach dem Supportende mit der auf ihrem Gerät installierten Software passiert.

6.3.4 Auswertung offener Fragen

In der Online-Befragung wurden drei offene Fragen gestellt, in denen die Befragten einen eigenen Text eingeben konnten. Diese drei Fragen lauteten:

- ▶ Was ist für Sie die wichtigste Eigenschaft eines vernetzten bzw. smarten Geräts?
- ▶ Wer trägt Ihrer Ansicht nach die Verantwortung für die Software auf Ihren Geräten und warum?
- ▶ Was sollte die Politik tun, damit smarte Geräte länger halten und nicht aufgrund von Software-Problemen den Geist aufgeben?

Die Antworten wurden qualitativ interpretiert und in sinnhafte Kategorien gruppiert. Die einzelnen Charakteristika wurden nachfolgend übergeordneten Antwortkategorien zugeordnet und sind ihrer Häufigkeit nach absteigend sortiert.

Zur Frage nach der für die Nutzer*innen wichtigsten Eigenschaft vernetzter / smarterer Geräte ergeben sich der Häufigkeit nach die Antwortkategorien:

1. Praktikabilität/ Nutzerfreundlichkeit und Produktivität,
2. Zuverlässigkeit/ Sicherheit/ Kontrolle
3. zeitliche sowie ökologische Nachhaltigkeit.

Für die meisten Befragten besteht die wichtigste Eigenschaft in einer einfachen Bedienung des Gerätes, auch unabhängig von einer smarten Funktion. Es folgen Zeit- und Ortsunabhängigkeit der Geräte. Letzteres gilt sowohl für Geräte, die innerhalb eines Hauses/ einer Wohnung bedient werden aber auch für die Steuerung der Geräte aus der Ferne (etwa während Nutzer*innen vereist sind) sowie Kompatibilität. Eine generelle Funktionalität ist wichtiger als eine smarte Funktionalität. Die zweithäufigste Antwortkategorie Zuverlässigkeit / Sicherheit / Kontrolle enthält hauptsächlich Äußerungen zu Störungsanfälligkeit, zeitlich unabhängige (dauerhafte) Verbindungen der Geräte zum Internet und die Datensicherheit. In einer dritten Antwortkategorie werden Wünsche nach Langlebigkeit sowie Nachhaltigkeit im Sinne von Energieeffizienz und Strahlungsarmut betont.

In den offenen Antworten wurden wahlweise Gerätebeschreibungen aber auch Wünsche für Geräteeigenschaften formuliert. Oft ist nicht klar auf was sich die Antworten beziehen. Insgesamt sind die Meinungen zu genutzten Geräten positiv; ein kleiner Teil der Befragten ist der Meinung bei den beurteilten Geräten existiere nur mangelhafte Datensicherheit.

Aus den Antworten kann zum Teil herausgelesen werden, dass vernetzte Geräte in ihrer Funktionsweise „schnell“ bzw. „schneller“ als nicht vernetzte Geräte eingeschätzt werden. Es ist unklar, ob sich die Geschwindigkeitseinschätzung auf das jeweilige Gerät oder die Netzgeschwindigkeit bezieht.

Der Preis der Geräte wird in den offenen Fragen nicht direkt thematisiert. In einigen Fällen wird jedoch auf Profitgier von Herstellern und Handel verwiesen, die in den Geräten „Sollbruchstellen“ einbauen, und für eine software- und hardwarebedingte kurze Haltbarkeit sorgen.

In Bezug auf die Verantwortung für die Software auf den jeweiligen Geräten ergeben sich vier Antwortkategorien:

1. Hersteller,
2. Befragte Person bzw. die Anwender*in selbst,
3. der Handel,
4. Gesetzgeber Politik

Die stark überwiegende Mehrheit der Befragten ist der Meinung, die Verantwortung für die Software liege beim Hersteller der Geräte. Ein geringer Teil der Befragten sieht Nutzer*innen der Geräte bzw. sich selbst als verantwortlich. Nur sehr wenige der Befragten nennen den Handel sowie Gesetzgeber/ Politik als für die Software verantwortlich.

Auch hier wird die Meinung vertreten, die zeitlich begrenzte Haltbarkeit der Geräte (software- oder hardwarebedingt) diene einer Profitmaximierung der Hersteller, weshalb diese auch für die Software-Haltbarkeit verantwortlich seien. Damit einhergehend werden Misstrauen und Wut gegenüber der Industrie ausgedrückt. Diese sei „unqualifiziert“, sowie „macht-, profit- und datenhungrig“.

Einige Befragten sehen sich selbst in der Verantwortung. Diesen Aussagen werden jedoch oft Vermerke hinzugefügt, die vermuten lassen, Verantwortung werde mit Handlungsmöglichkeit/ Handlungsfreiheit gleichgesetzt. So sind Befragte in einigen Fällen der Meinung, über Software-

Updates selbst entscheiden zu können, was jedoch lediglich die tatsächliche Installation und nicht die Funktionssteuerung der Geräte durch installierte Software betrifft.

In Bezug auf die zweite Antwortkategorie wird oft die Meinung vertreten, dass die Geräte den Nutzer*innen gehörten, sobald sie diese käuflich erworben hätten. Aus diesem Grund seien sie auch verantwortlich für die Software. Dabei wird anscheinend oft nicht zwischen Gerät und Software unterschieden.

In Bezug auf Handlungsmöglichkeiten der Politik finden sich drei hauptsächliche Antwortkategorien:

1. Gesetze erlassen/ Garantieverlängerungen (für Hard- und Software) durchsetzen/ Kompatibilität erhöhen
2. die Politik kann nichts tun / die Politik solle nichts tun
3. mehr Ressourcen in Bildung und Forschung zu investieren, um Geräte länger haltbar zu machen

Die erste Antwortkategorie beinhaltet verschiedene konkrete Forderungen an die Politik. Bezüglich der Geräte und der von Herstellern installierten Software sollen Hersteller verpflichtet werden, langlebige und reparierbare Produkte zu produzieren und die Abwärtskompatibilität bei Software- und Hardware zu erhöhen. Diese Forderungen sollen durch Anreize und Prämien unterstrichen werden. Eine weitere Forderung in der ersten Antwortkategorie bezieht sich auf das Erlassen von Verboten und Strafen (Sanktionierung) von Herstellern bei Updateabhängigkeit, ein Verbot fest verbauter Akkus sowie eine Entsorgungsverpflichtung der Hersteller. Eine dritte Forderung an die Politik betrifft aktive Aufklärung zu Produkthaltbarkeiten und deren (nicht vorhandene) Wiederverwertbarkeit.

In einer zweiten Antwortkategorie sind ein Großteil der Befragten der Meinung, dass die Politik nichts an der Situation ändern kann, oder auch nicht will, da sie ein „Spielball der Wirtschaft“ sei und „selbst von der Kurzlebigkeit der Produkte profitiere“. Einige Befragte sind der Meinung, dass sich die Langlebigkeit bestimmter Produkte durch Mundpropaganda herumsprache und damit weniger haltbare Produkte vom Markt verdrängt würden. Den Verbraucher*innen werden also ein hoher Einfluss und mit dem eigenen Konsumverhalten auch eine hohe Verantwortung beigemessen, auf Produktcharakteristika einzuwirken.

Eine dritte, weniger häufig vorkommende Antwortkategorie beinhaltet die Forderung an die Politik, mehr Ressourcen in Bildung und Forschung zu investieren. Dies bezieht sich zum einen darauf, über globale Rohstoffknappheit oder gesundheitliche Schädlichkeit verbauter Materialien zu informieren und zum anderen darauf, auch bei der Entwicklung und Produktion neuer Geräte diese Aspekte mit in Betracht zu ziehen.

6.4 Fazit

Daten zu Nutzer*innen-Erfahrungen mit softwarebedingter Obsoleszenz waren vor dieser Studie noch nicht systematisch erhoben worden. Im Rahmen des Projekts wurden diese mittels unterschiedlicher Methoden untersucht: einerseits wurden Produktrezensionen und Foreneinträge von Nutzer*innen im Internet auf Inhalte mit Bezug zu Software-Obsoleszenz untersucht. Außerdem wurden 1000 repräsentativ ausgewählte Menschen zwischen 18 und 75 Jahre mit einer Online-Befragung direkt zu aktuell im Haushalt eingesetzten smarten Elektronikprodukten, geplanten Anschaffungen, wahrgenommenen Verschlechterungen, Praktiken im Umgang mit Problemen und Erwartungen an Produkte befragt.

Ergebnisse der qualitativen Auswertung von Produktrezensionen und Onlineforen

- ▶ Smarte Geräte, die nur in Verbindung mit Server-Infrastruktur funktionieren oder auf Smartphone-Apps zur Steuerung angewiesen sind, können spontan unbenutzbar werden, in Einzelfällen bereits kurz nach dem Kauf
- ▶ Hersteller kommunizieren häufig technische Gründe für softwarebedingte Obsoleszenz, die von außen nicht überprüfbar und für Nutzer*innen unbefriedigend sind
- ▶ Smarte Geräte funktionieren oft besonders gut im Zusammenspiel mit anderen Geräten desselben Herstellers. Dadurch können Lock-In-Effekte entstehen und es droht die Obsoleszenz einer großen Anzahl von Geräten im Haushalt, wenn sich Schnittstellen verändern
- ▶ Für Details siehe Kapitel 6.2

Ergebnisse der Onlinebefragung - geschlossene Fragen

- ▶ Smartphones, Internet-Router, PCs und Smart-TVs werden in der Mehrzahl der Haushalte genutzt
- ▶ Smart-TVs, Saugroboter, Beleuchtung und Sicherheitsprodukte sind die am häufigsten geplanten Anschaffungen smarterer Geräte
- ▶ Die meisten Befragten sind mit den genutzten smarten Geräten zufrieden
- ▶ Jüngere Befragte nehmen Verschlechterungen smarterer Geräte über die Benutzungszeit häufiger wahr als ältere Befragte
- ▶ Während jüngere Nutzer*innen eher online nach Lösungen für Probleme mit Geräten suchen, fragen ältere Nutzer*innen eher Familienmitglieder und Bekannte
- ▶ Eine Mehrheit der Befragten wünscht mehr Informationen zur Haltbarkeit und zur Software-Unterstützung von Geräten, mehr Wissen zur sachgemäßen Pflege und Wartung, mehr Unterstützung bei Fehlern und Einschränkungen und mehr Informationen durch den Handel und die Hersteller
- ▶ Ca 50 % der Befragten wünscht mehr Rechte für Nutzer*innen und mehr Informationen über die Umweltbelastungen

Ergebnisse der Onlinebefragung - offene Fragen

- ▶ wichtigste Eigenschaften vernetzter Geräte aus Sicht der Nutzer*innen
 - 1) Praktikabilität/ Nutzerfreundlichkeit und Produktivität
 - 2) Zuverlässigkeit/ Sicherheit/ Kontrolle
 - 3) zeitliche sowie ökologische Nachhaltigkeit
- ▶ Für die überwiegende Mehrheit der Befragten liegt die Verantwortung für die Software beim Hersteller der Geräte. Ein geringer Teil der Befragten sieht sich selbst als Nutzer*in verantwortlich. Handel sowie Gesetzgeber/ Politik werden von wenigen genannt.
- ▶ Handlungsmöglichkeiten der Politik nach Häufigkeit
 - 1) Gesetze erlassen / Mindestnutzungszeiträume definieren / Kompatibilität erhöhen
 - 2) keine politischen Handlungsmöglichkeiten gegeben / Politik soll sich nicht einmischen
 - 3) mehr in Bildung und Forschung investieren, um Geräte länger haltbar zu machen

7 Wahrnehmungen und Erfahrungen von Expert*innen und Praktiker*innen

7.1 Fragestellungen und Vorgehensweise

7.1.1 Begleitkreis

Da das Thema softwarebedingte Obsoleszenz verschiedene Bereiche der Gesellschaft betrifft und ein ganzheitlicher Ansatz zur Lösung dieser Herausforderung nötig ist, wurden im Rahmen des Vorhabens unterschiedliche Akteursgruppen über die Einrichtung eines Begleitkreises einbezogen. Ziel der Begleitkreistreffen war die gemeinsame Diskussion mit Expert*innen, die ihre spezifischen Erfahrungen und Kenntnisse in die Erarbeitung der Ergebnisse einbringen sollen.

Der Begleitkreis setzt sich zusammen aus Expert*innen verschiedener gesellschaftlicher Bereiche und Sektoren (Forschung, Unternehmen und Start-Ups im Software- und Elektronik-Bereich, Branchenverbände, Vertreter*innen aus Zivilgesellschaft und Open Source-Bewegung).

7.1.2 Akteursinterviews

Im Februar 2021 wurden acht Akteursinterviews mit Vertreter*innen von Herstellern, Verbraucherinitiativen, Verbänden und Interessensgruppen geführt. Ziel der Interviews war es, den Stand des Wissens zu softwarebasierter Obsoleszenz zu erfassen – einerseits zu Ursachen und Entstehung von SobO und andererseits zu Einflussmöglichkeiten auf SobO von Seiten der Hersteller, Verbraucher*innen und der Politik.

Für die Interviews wurde ein einheitlicher Gesprächsleitfaden genutzt, der im Anhang A dieses Dokuments zu finden ist. Die 30- bis 60-minütigen Gespräche wurden aufgezeichnet, transkribiert und deskriptiv ausgewertet.

7.2 Ergebnisse der Begleitkreis-Workshops

7.2.1 Problemwahrnehmung und Herausforderungen

These 1: Für Konsument*innen ist softwarebedingte Obsoleszenz gegenwärtig noch kein Problem, die Markttrends weisen jedoch auf eine Vielzahl zukünftiger Problematiken hin.

- ▶ Empirische Fälle für SobO sind (bisher) überschaubar
- ▶ Umfragen zeigen, dass bisher andere Faktoren bei der Obsoleszenz von Produkten überwiegen
- ▶ In der Forschung wird das Problem im Bereich der Konsumerelektronik bisher nicht diskutiert
- ▶ Die Anzahl softwarebasierter und vernetzter Geräte in den Haushalten nimmt zu, die daraus resultierenden Obsoleszenzrisiken steigen

Ergebnisse der Diskussion während des Workshops

Die Beobachtung, dass der Anteil von Software an der Wertschöpfung von Produkten wächst, wurde bestätigt. Die hohe Komplexität von Software ist dabei eine der zentralen Herausforderungen bei der Entwicklung, konkurrierende Standards führen zu Kompatibilitätsproblemen. Softwarebasierte Produkte unterscheiden sich von anderen Produkten, da sie in der Nutzungsphase verändert werden können – auch durch Dritte. Daher ist

ein besonderer Fokus auf Updates notwendig. Quelloffene Software kann für Hersteller und Nutzer*innen Möglichkeiten bieten, Obsoleszenz zu vermeiden.

These 2: Die Anbieter von smarten Geräten haben aufgrund der starken Verflechtungen zwischen Soft- und Hardware-Zulieferern sowie Dienstleistern wenig Handlungsspielräume, um lange Lebensdauern zu gewährleisten.

Ergebnisse der Diskussion während des Workshops

Es existieren unterschiedliche Ansätze, um die Nutzungsphase von Geräten zu verlängern: existierende Geräte können zum Beispiel einer Zweitnutzung zugeführt werden. Neue Geräte sollten bereits in der Designphase auf Langlebigkeit ausgelegt werden. Dazu wurden aus dem Bereich Software verschiedene Konzepte angesprochen: modulare Software könnte besser auf die Bedürfnisse der Nutzenden zugeschnitten sein und daher die Leistung der Geräte besser ausnutzen. Andererseits schafft dies wiederum Komplexität, die Qualitätsmanagement erschwert. Gut definierte, herstelleroffene Schnittstellen können Obsoleszenz verhindern, derzeit gibt es allerdings oft konkurrierende Standards oder geschlossene Ökosysteme.

7.2.2 Lösungsstrategien

Ergebnisse der Breakoutgruppen

- ▶ Frage: Was können wirtschaftliche Akteure tun?

Sie sollten Langlebigkeit und Software-Effizienz als Anforderungen an die zu entwickelnden Produkte definieren und von Beginn an verfolgen. Enger Austausch mit den Nutzer*innen über das gesamte Produktleben kann einen Wissenstransfer in alle Richtungen ermöglichen und die Mündigkeit der Verbraucher*innen stärken.

- ▶ Frage: Welchen Beitrag können zivilgesellschaftliche Akteure leisten?

Sie können Aufmerksamkeit für den Themenkomplex generieren, Zielkonflikte und Dilemmata ansprechen und gute Praktiken der Produktentwicklung dokumentieren. Bei der Aushandlung politischer Maßnahmen sollten sie die Bedürfnisse der vertretenen Stakeholder kommunizieren.

- ▶ Frage: Welche Allianzen könnten für Strategien gegen Softwareobsoleszenz hilfreich sein?

Gut dokumentierte, offene Standards können die Interoperabilität und Produktlebensdauern erhöhen. Durch Tagungen, Hackathons und Workshops sollten Gelegenheiten zum Austausch über Nachhaltigkeitsthemen geboten werden.

Ideen und Vorschläge für die weitere Arbeit des Projektteams

- ▶ Prinzipien und Möglichkeiten der Gestaltung „schlanker“ sowie „modularer Software“
- ▶ Möglichkeiten der Erarbeitung ethischer Standards für „suffiziente Software“
- ▶ Verbesserung der Transparenz von Software-Unterstützung (Dauer, Art, funktionserhaltende vs. funktionserweiternde Updates) gegenüber Nutzer*innen
- ▶ Möglichkeiten der Integration firmenfremder Software sowie der Anschlussnutzung mit Open Source-Software

- ▶ Vertiefte Betrachtung von Zielkonflikten, z.B. zwischen großen Plattformbetreibern und Hardware-Herstellern
- ▶ Berücksichtigung der relativen Kontrolle und Einflussmöglichkeiten der verschiedenen Akteure in der Lieferkette
- ▶ Identifikation destruktiver Praktiken: wie wird Software ggf. gezielt eingesetzt, um Lebensdauern zu verkürzen?

7.3 Ergebnisse der Interviews

7.3.1 Problemwahrnehmung und Herausforderungen

Die Befragten sind sich einig, dass Software zunehmenden Einfluss auf die Nutzungsdauer von Geräten hat. Innerhalb der eigenen Branchen wird von den Befragten ein verstärktes, wenn auch oft noch geringes Interesse an dem Thema softwarebasierte Obsoleszenz wahrgenommen. Weiterhin stimmen sie darin überein, dass Nutzer*innen, Hersteller, Vereine und Verbände sowie die Politik gefordert seien, Anstrengungen zu unternehmen, um langlebige Produkte mit Softwareanteil zu fördern und zu nutzen. Tendenziell wird Software von den Befragten eher als Risiko für die Nutzungsdauer gesehen, insbesondere bei Geräten, die bisher ohne Softwareanteil realisiert wurden. Daneben wurde aber auch erwähnt, dass Software im besten Fall durch die Möglichkeit funktionaler Updates auch eine Verlängerung der Produktlebens durch eine Anpassung an neue Anforderungen ermöglichen könne.

Verschiedene Kontexte

Alle Befragten sind beruflich mit Software oder Elektronik befasst, allerdings in unterschiedlichen Branchen und auf unterschiedlichen Ebenen. Folgende *Kontexte* von softwarebasierten Produkten wurden angesprochen, teils aus professioneller, teils auch aus privater Perspektive:

Klassische Haushaltsgeräte und Peripherie-Geräte: Diese Produkte enthalten Software, sind aber typischerweise nicht vernetzt und bieten oftmals keine zugänglichen und dokumentierten Schnittstellen für Updates der Software. Es werden meist einfache Mikrocontroller eingesetzt, die wenig Speicher und geringe Rechenleistung aufweisen. Typischerweise ist ein Softwarestand über viele Jahre aktuell und es werden keine oder wenige Funktions- und Sicherheitsupdates bereitgestellt. Im Rahmen der Befragung wurden für diesen Bereich vor allem die Themen Software als Ersatzteil und Recht auf Reparatur genannt. Beispiele: Waschmaschine, Küchengeräte, PC-Peripherie wie Webcams, Mikrofone.

Smarte Geräte im Haushalt: Diese Produktgruppe wächst derzeit schnell. Im Vergleich zur ersten Gruppe sind die eingesetzten Chips tendenziell leistungsstärker. Die Produkte entfalten ihre Funktionalität in Gänze erst im Zusammenspiel mit anderen Geräten und sind oft an das Internet angeschlossen. Durch die stetige Veränderung des Umfelds der Geräte, kann eine gleichbleibende Dienstqualität nur durch regelmäßige Updates erreicht werden. Dabei wird von mehreren Befragten zwischen Sicherheits- und Funktionsupdates unterschieden. Sicherheitsupdates sind insbesondere bei Geräten mit Internetverbindung notwendig, da die dauerhafte Sicherheit der Produkte ohne Softwareupdates nicht sicherzustellen ist. Die Notwendigkeit von Updates ist dabei nach Ansicht der Befragten noch nicht allen Anbietern und Verbraucher*innen ausreichend bewusst.

Im Rahmen der Befragung wurden für diesen Bereich die folgenden Themen angesprochen: Software (SW) als Ersatzteil, Mindesthaltbarkeit von SW, Komplexität des Softwarestacks,

Offenlegung des Quellcodes nach Supportende, mangelndes Bewusstsein für SW-Qualität bei Anbietenden und Endverbraucher*innen, SW-Qualität als Organisationsthema, Open Source.

Smartphones: diese Produktklasse kam in allen Interviews zur Sprache. Zusätzlich zu den Eigenschaften der smarten Geräte im Haushalt wurden hier folgende Punkte angesprochen: Die sehr große Anzahl der Geräte auf dem Markt mit unterschiedliche Hardware- und Softwareständen und die schnelle Ablösung durch Nachfolgemodelle machen dauerhaften Support teuer für die Hersteller und Entwickelnden von Apps. Verbraucher*innen und Hersteller haben sich an eine Nutzungsdauer von zwei bis drei Jahren „gewöhnt“. Die Leistungsfähigkeit der Hardware steigt schnell, dadurch steigen auch die HW-Anforderungen der Software, diese ist dann auf älteren Geräten nicht mehr gut einsetzbar. Genannte Themen: Mindesthaltbarkeit von Software, Komplexität des Softwarestacks, Offenlegung des Quellcodes nach Supportende, mangelndes Bewusstsein für SW-Qualität bei Endverbraucher*innen, Gerätepfand, Produktlebenszyklus, Open Source.

Software für general purpose Computer: in mehreren Interviews wurde Software zur Installation auf Desktop-Rechnern angesprochen. Dabei ging es wiederum um die Leistungsanforderungen der SW an die Hardware. Im Vergleich zu den anderen Bereichen wurde hier vermehrt auch auf ökologische Vorteile effizienter Software mit niedrigerem Energieverbrauch hingewiesen und der Blaue Engel für effiziente Software erwähnt. Im Vergleich zu Smartphones verlaufen die Leistungszunahme und der Technologiewandel der Hardware im Desktop-Bereich langsamer. Wichtig ist hier die regelmäßige Anpassung der Software an die Betriebssysteme und deren Sicherheitsinfrastruktur. Große SW-Anbieter versuchten gezielt, ihre Produkte als defacto-Standard zu etablieren und durch proprietäre Datenformate die Kompatibilität zu Konkurrenz- und Open-Source-Anbietern einzuschränken.

Genannte Themen: Mindesthaltbarkeit von Software, Komplexität des Softwarestacks, Offenlegung des Quellcodes nach Supportende, SW-Geschäftsmodelle, Produktlebenszyklus, Open Source.

Software und Reparatur

Die Befragten sind sich einig, dass der zunehmende SW-Anteil in Elektronikgeräten eine Herausforderung für deren Reparatur darstellt. Elektronische Bauteile können typischerweise im Schadensfall identifiziert und durch typengleiche Ersatzteile, vom Originalhersteller oder von anderen, ersetzt werden. Sind jedoch programmierbare Bauteile betroffen, so muss die darauf ausgeführte SW entweder vom Hersteller bereitgestellt oder erneut geschrieben (re-implementiert) werden. Die Re-implementierung erfordert aber genaues Wissen über das System und ist aufwendig, für einzelne Reparaturen typischerweise nicht wirtschaftlich. Hier ist aus Sicht der Befragten ein Verständnis der Hersteller erforderlich, dass auch und insbesondere SW und ihre Dokumentation als Ersatzteil zur Verfügung gestellt werden müssen.

Einen Spezialfall in diesem Bereich stellt SW dar, die gezielt die Reparatur von Geräten durch Dritte verhindern soll. Bisher sind solche Fälle bei Smartphones und Haushaltsgeräten bekannt, aber noch selten. Alle Befragten sind sich einig, dass SW nicht zur Verhinderung einer wirtschaftlichen Reparatur eingesetzt werden soll.

Im Detail unterscheiden sich die Ansichten, ob eine Einschränkung des Zugriffs auf Ersatzteile und Dokumentationsmaterial auf professionelle Reparaturbetriebe sinnvoll sein könnte für Reparaturen, die die Produktsicherheit betreffen. Die Veränderung von Steuerungssoftware kann die Eigenschaften der Geräte verändern, eine veränderte Ansteuerung von Sensoren und Aktoren kann zur Überlastung von Komponenten und Versagen der Geräte, auch mit negativen Auswirkungen auf Menschen und Umwelt führen.

Komplexität und Softwarestack

Software wird oft durch einen komplexen Softwarestack realisiert, d.h. die SW wird nicht zentral koordiniert entwickelt, sondern entsteht aus verschiedenen *Schichten* und unter Rückgriff auf existierende SW-Bibliotheken von Dritten. Ziel dieses modularen Aufbaus ist die Wiederverwendbarkeit von SW-Modulen für verschiedene Projekte. Die unterschiedlichen SW-Module und Komponenten kommunizieren bestenfalls nur über wohldefinierte Schnittstellen und sind ansonsten entkoppelt.

Gerätehersteller haben dadurch meist keinen Zugriff auf den Quellcode aller verwendeter SW und verfügen auch nicht über die technischen Fähigkeiten, um alle Schichten des SW-Stacks zu verstehen und zu warten. Daher sind sie im Fall von Sicherheitslücken oder zur Implementierung zusätzlicher Funktionen auf Zulieferer angewiesen. Die unteren Schichten des SW-Stacks müssen genau auf die verwendete Hardware angepasst sein und werden typischerweise vom Zulieferer der Chips bereitgestellt. Dies verhindert in manchen Bereichen ein Update auf neuere Softwarestände in oberen Schichten des SW-Stacks, z.B. bei Smart-TV-Geräten und auch Smartphones, da die Schnittstellen zwischen den Schichten nicht stabil sind.

SW-Geschäftsmodelle

Verschiedene Geschäftsmodelle für Software sind verbreitet und führen zu unterschiedlichen Anreizen in Bezug auf Obsoleszenz: Software, die einmal bezahlt wird, Software mit Kauf und anschließenden kostenpflichtigen Supportverträgen, Abo-Modelle und Cloud-Angebote, bei denen die Prozesse auf Rechenzentren des SW-Anbieters ausgelagert sind. Die Geschäftsmodelle wurden in den Interviews sowohl aus Nutzer*innen- als auch aus Anbieter-Sicht betrachtet, im Mittelpunkt der Betrachtungen stand dabei SW für Desktop-Computer und Smartphones, die generellen Aussagen dürften aber auf die anderen Produktgruppen übertragbar sein. Aus Nutzer*innen-Sicht wird insbesondere das Obsoleszenzrisiko von Spezialsoftware als gefährlich gesehen, die für die eigene Arbeit essenziell ist: Abkündigungen können die eigene Arbeitsfähigkeit gefährden, Preiserhöhungen müssen im Zweifel akzeptiert werden. Cloud- und Abomodelle schaffen zusätzliche Abhängigkeiten. Open-Source-Software oder Eigenentwicklungen bieten hier Vorteile., Mehrere Befragte sehen die Verringerung von Abhängigkeiten als zentrales Kriterium für IT-Entscheidungen.

Aus Anbietersicht wurden als Gründe für erfolgte Abkündigungen technische Hürden für einen weiteren Support auf der einen und betriebswirtschaftliche Erwägungen am Ende des Produktlebenszyklus auf der anderen Seite genannt: während die Zahl der Nutzenden in dieser Phase abnimmt, steigt der Aufwand für Updates durch den großen zeitlichen und technologischen Abstand zur ursprünglichen Programmierung. Abo- und Supportvertragsmodelle, die Umsatz entsprechend der realen Nutzung der Software auch in späteren Phasen des Lebenszyklus generieren, werden aus Anbietersicht tendenziell begrüßt. Insbesondere ein Befragter sieht im Ausbau von Cloud-Angeboten großes Potential, um mit einfachen, langlebigen Geräten auf Seite der Benutzenden eine stets aktuelle, funktionale und sichere Software anbieten zu können, die zentral gewartet, verbessert und nutzungsabhängig bezahlt wird. Von anderen Befragten wird dieses Modell mit Verweis auf die eingeschränkte Souveränität, Datenschutzaspekte und das Abhängigkeitsverhältnis gegenüber Großkonzernen abgelehnt.

7.3.2 Unterschiede in den Wahrnehmungen

Die Befragten sind sich einig, dass Handlungsbedarf in Hinsicht auf softwarebedingte Obsoleszenz besteht und haben eine gemeinsame, jedoch unscharfe Vorstellung, was gute Software ausmacht. Bereits bei der Formulierung konkreter Ziele zeigen sich Unterschiede:

insbesondere Befragte, die beruflich selbst mit der Entwicklung von Software oder Elektronikgeräten befasst sind, weisen auf den direkten Zusammenhang zwischen geforderter Qualität – auch in Hinsicht auf Obsoleszenzrisiken – und Entwicklungskosten hin. Eine deutliche Verminderung des Risikos softwarebedingter Obsoleszenz sei möglich, aber mit höheren Entwicklungskosten verbunden. Dies könnte negative Auswirkungen auf die Wettbewerbsfähigkeit der Produkte haben. Es brauche daher einen Aushandlungsprozess, um das Optimum zwischen Entwicklungsaufwand und Minimierung des Obsoleszenzrisikos zu erreichen.

Auch die Erfolgsaussichten staatlicher Regulierung zum Finden und Erreichen dieses Optimums werden sehr unterschiedlich bewertet. Insbesondere Entwickler*innen äußern Bedenken zur konkreten Ausgestaltung von Regelungen in der eigenen Domäne und befürchten zusätzlichen Aufwand, dem kein entsprechender individueller und gesellschaftlicher Nutzen entgegensteht.

Andere Befragte stehen regulatorischen Maßnahmen deutlich positiver gegenüber und sehen gute Erfolgsaussichten.

Das Thema Geschäftsmodelle wurde in fast allen Interviews angesprochen. Meist ging es dabei um SW, die auf general purpose Computern installiert wird, die Aussagen lassen sich aber zum Teil auf andere SW und Geräte mit großem SW-Anteil übertragen. Bei diesem Thema zeigen sich viele Spezifika von SW wie die völlige Veränderbarkeit durch Updates, die komplexe Zulieferkette der Software-Stacks, Abhängigkeit von Schnittstellen und Servern. Daraus folgen neue rechtliche Fragestellungen im Blick auf Eigentum und Verantwortung für Ausfälle und Schäden. Obsoleszenzrisiko, betriebswirtschaftliche und rechtliche Risiken sowie Safety- und Security-Risiken stehen in einem Spannungs- und Abhängigkeitsverhältnis.

7.3.3 Lösungsstrategien

Den unmittelbaren Einfluss der Verbraucher*innen auf das Obsoleszenzrisiko der angebotenen Geräte und Software halten die Befragten für gering. Mittelbar könnten Nutzer*innen aber durch Fragen nach Informationen zu Lebensdauer und Nachfragen zu Versprechen der Hersteller im Verkaufsgespräch oder in Kontakt mit den Kundenservices Druck aufbauen und dadurch die Anforderungen für neu zu entwickelnde Produkte verändern. Viele Befragte wünschen sich ein größeres Bewusstsein der Nutzer*innen für die Lebensdauerkosten: langlebige Produkte seien oft teurer in der Anschaffung und günstiger im Unterhalt, Konsumententscheidungen würden aber stark vom Verkaufspreis geleitet. Verbraucherschutzorganisationen könnten durch Kampagnen Aufmerksamkeit auf das Thema softwarebedingte Obsoleszenz lenken und Konsument*innen zu verantwortungsvollen Kaufentscheidungen befähigen.

Die Hersteller haben einen großen Einfluss auf das Obsoleszenzrisiko der entwickelten Produkte. Die Befragten sind sich einig, dass SW-Qualität ein Organisationsthema ist, das nicht von einzelnen Entwickler*innen im Alleingang angegangen werden kann. Mindest-Produktlebensdauern sollten als Anforderungen zu Beginn des Entwicklungsprozesses definiert werden. Das Zielbild „gute SW“ sollte gesetzt und mit Inhalt gefüllt werden, dabei besteht aber unter den Befragten über abstrakte Schlagworte hinaus keine Einigkeit, wie „gute SW“ zu erreichen wäre. Genannte Schlagworte sind „schlanke SW“, „offene / dokumentierte Schnittstellen“, „Modularität (unterschiedliche Bedeutung je nach Problemwelt) / Entkopplung / Abstraktionslayer (eher bei komplexer SW)“. Verbände und NGOs können bei der Etablierung geeigneter Entwicklungsprozesse beraten und werden bereits verstärkt aus der Industrie angesprochen.

Unterschiedliche Maßnahmen der Politik werden von den Befragten genannt und auch in der Bewertung der Maßnahmen besteht keine Einigkeit. Im Folgenden werden die am häufigsten angesprochenen Maßnahmen und ihre unterschiedlichen Bewertungen kurz vorgestellt.

Tabelle 16: Im Interview vorgeschlagene Maßnahmen

Maßnahme	Potenziale	Risiken / Schwierigkeiten / Uneinigkeit
Recht auf Reparatur	Wird generell begrüßt, breite Zustimmung unter den Befragten. Insbesondere SW, die Reparatur gezielt verhindert, wird von den Befragten abgelehnt.	Welche Rolle spielt SW für die Reparatur? SW als Ersatzteil, Risiken für Mensch und Umwelt durch Softwaremanipulation, Beschränkung auf Fachbetriebe?
Siegel für gute Produkte	Bessere Erkennbarkeit für Verbraucher*innen	Bereits viele Siegel mit unterschiedlicher Bekanntheit und Aussagekraft vorhanden
Verlängerte Gewährleistungsfristen / Mindestzeiträume für die Versorgung mit Updates	Längere Produktlebensdauern, tendenziell verringerte Produktvielfalt mit positiven Auswirkungen auf Ersatzteilversorgung und tatsächliche Reparierbarkeit	Manche Branchen (z.B. Smartphones) mit sehr kurzen Releasezyklen und hoher SW-Komplexität. Hersteller sind auf Zulieferer angewiesen, großer Aufwand für Hersteller
Mindestlebensdauern für SW	Längere und planbare Einsatzzeiträume	SW-Hersteller haben keine Kontrolle über die HW, unklare Verantwortlichkeiten
Verpflichtende Offenlegung des Quellcodes nach Supportende	Community oder Drittanbieter könnten SW pflegen und so die Lebensdauer verlängern, neue Features für ältere Hardware	Rechtliche Fragen: „was ist mit zugeliefertem Quellcode / Bibliotheken?“, Know-How oft im Quellcode, würde der Konkurrenz zugänglich werden. Auswirkungen auf Geschäftsmodelle; Haftung für das Verhalten von HW nach SW-Änderungen durch Dritte
Gerätepfand	Eher geringes Potenzial im Hinblick auf softwarebedingte Obsoleszenz	

Quelle: Eigene Darstellung (Institut für Sozialinnovation Consulting UG/ TU Berlin)

7.4 Zusammenfassung und Fazit

Das Thema softwarebedingte Obsoleszenz wird von den unterschiedlichen Akteuren im Bereich IKT als zunehmend relevant wahrgenommen und bearbeitet. Dabei gibt es bisher keine allgemein genutzte Definition des Begriffes softwarebedingte Obsoleszenz, so dass unterschiedliche Mechanismen und Vorgänge darunter verstanden werden.

Einigkeit besteht darüber, dass Obsoleszenzrisiken durch geeignetes Vorgehen während Produktplanung, -entwicklung und Support verringert werden könnten. Die Vermeidung von softwarebedingter Obsoleszenz liegt nicht in der Hand Einzelner, sondern muss als Organisationsthema verstanden werden.

Gerade bei Consumer-Produkten stehen die unmittelbaren Kosten beim Erwerb bei Kaufentscheidungen im Vordergrund, während die Lebenslaufkosten weniger betrachtet werden. Sorgfältig entwickelte Produkte mit geringerem Obsoleszenzrisiko und erhöhter Haltbarkeit sind aber typischerweise mit höheren Anschaffungskosten verbunden und erst bei Betrachtung der längeren möglichen Nutzungsdauer preislich interessant.

Mögliche Siegel für gute Produkte, gesetzliche Mindestlebensdauern für Produktklassen, Verbesserungen der Reparierbarkeit von Produkten mit Softwareanteil, Mietmodelle als Alternative zum Gerätekauf und eine Offenlegungspflicht von Quellcode nach Supportende wurden von den Befragten angesprochen. Über konkrete Vorgehensweisen und geeignete politische Anreize, die zu *guter Software* mit geringem Obsoleszenzrisiko führten, besteht jedoch keine Einigkeit.

8 Ableitung produktpolitischer Maßnahmen

8.1 Zielsetzung

Die vorangehenden Untersuchungen haben deutlich gemacht, dass Software einen wesentlichen Einfluss auf die Nutzungsdauer von vernetzten und mikroelektronischen Produkten hat. Dieser Einfluss kann positiv sein, indem Software neue Funktionalitäten in vorhandene Produkte integriert und sie dadurch länger nutzbar macht. Der Einfluss kann aber auch negativ sein, indem die Software geänderte Anforderungen nicht mehr erfüllt (z.B. keine Unterstützung neuer Übertragungs- oder Sicherheitsstandards), indem die Software zu viele Hardware-Ressourcen in Anspruch nimmt (z.B. aufgeblähte Software-Updates) oder indem die Software externe Abhängigkeiten schafft, die nach einer gewissen Zeit nicht mehr erfüllt werden (z.B. abgestellte Cloud-Services oder Lizenz-Server). Wirkt Software negativ auf die Nutzungsdauer von Produkten, so kann von softwarebedingter Obsoleszenz gesprochen werden. Produkte werden wegen der Software vorzeitig obsolet, obwohl ihre Hardware noch funktionstüchtig ist.

Softwarebedingte Obsoleszenz führt zu einem erhöhten Aufkommen an Elektronikschrott, einem erhöhten Ressourcenverbrauch zur Herstellung von Neugeräten und zu wirtschaftlichen Belastungen für die Anwender*innen bzw. Konsument*innen solcher Produkte. Einer softwarebedingten Obsoleszenz sollte deshalb politisch entgegengesteuert werden. Ziel von produktpolitischen Maßnahmen ist es in diesem Fall, eine möglichst lange Nutzungsdauer von Produkten zu ermöglichen. Die Maßnahmen sollten sowohl die Hersteller und Händler von Produkten adressieren, aber auch die für die Funktionalität der Produkte erforderlichen Dienstleister (z.B. Cloud-Service-Anbieter, Software-Entwickler) und die Verbraucher*innen selbst, die letztlich über die Entsorgung und den Neukauf von Produkten entscheiden.

Nachfolgend werden verschiedene regulatorische Ansatzpunkte für eine Produktpolitik untersucht, die der softwarebedingten Obsoleszenz entgegenwirken könnten. Anschließend werden daraus Handlungsempfehlungen abgeleitet, welche produktpolitischen Maßnahmen in Regulierungen umgesetzt werden sollten und damit zu einer nachhaltigen Markttransformation beitragen können.

8.2 Regulatorische Ansatzpunkte

In Bezug auf direkte und indirekte Umwelt- und Nachhaltigkeitsanforderungen an Software gibt es bisher nur wenige oder gar keine Standards und Regulierungen (Gröger & Köhn 2015; Hilty et al. 2015). Software spielt innerhalb der integrierten Produktpolitik der EU (IPP) eine bisher untergeordnete Rolle, die im Zuge der Digitalisierung und Verbreitung softwarebasierter Systeme und Produkte jedoch zunehmend an Bedeutung gewinnt. Politische Maßnahmen und Normierungen erfolgen auf verschiedenen Ebenen und unterscheiden sich in ihrem Grad der Verbindlichkeit. Da es keine übergreifende Regelung oder allumfassenden Standard für die Regulierung von Software und die mit ihr verbundenen Wirkungen gibt, soll an dieser Stelle dennoch auf eine Reihe von wegweisenden europäischen und nationalen Beschlüssen sowie Forderungen eingegangen werden, von den in der Praxis eine normierende Wirkung („Soft Laws“¹⁶) ausgeht, deren Einhaltung an dieser Stelle jedoch nicht weiter überprüft werden soll. Zuzüglich wird auf die bestehenden zivilrechtlichen Möglichkeiten und Entwicklungen („Hard Law“) eingegangen.

¹⁶ Unter „Soft Laws“ werden nicht verbindliche Übereinkünfte, Absichtserklärungen, Leitlinien und Standards verstanden, die gegenüber den „Hard Laws“ eine geringere Selbstbindung aller Akteure und Akteurinnen aufweisen. Damit sind diese jedoch nicht wirkungslos, sondern können durch Selbstverpflichtung und Quasi-Standards normierend wirken.

Mit der wachsenden öffentlichen Aufmerksamkeit um das Thema der geplanten Obsoleszenz, der Forderung nach mehr Haltbarkeit und längeren Produktlebensdauern sowie den anhaltenden Debatten zur nachhaltigen Ausgestaltung der Digitalisierung ist das Thema der softwarebedingten Obsoleszenz in den letzten Jahren auf die politische Agenda gekommen und wird in Deutschland insbesondere innerhalb des Umwelt- und Klimaschutz politisch adressiert (siehe Tabelle 17).

Tabelle 17: Übersicht zu wesentlichen politischen Entwicklungen und Forderungen mit Bezug zu Softwareobsoleszenz (nach Erscheinungsjahr geordnet)

Bezeichnung	Quelle	Policy-Format	Bezüge zu Softwareobsoleszenz
Circular Economy Action Plan (CEAP)	EC (2020b)	Zielkatalog	Recht auf Updates von obsoleter Software
Umweltpolitische Digitalagenda des BMU (2020)	BMU (2020)	Maßnahmen und Zielkatalog	Forderung für verbindliche Herstellervorgaben in der EU-Ökodesign-Richtlinie, für die lange Funktionsfähigkeit von Hardware-Software-Systemen, Verbindliche Regelung zur Bereitstellung von Updates als Teil einer Garantieaussagepflicht und eines Rechts auf Reparatur, Förderung von „Grünem Coding“ und Methoden der ressourcenschonenden Softwareentwicklung
Herstellieranforderungen zur Sicherheit von Smartphones des BSI (2020)	BSI (2020a)	Unverbindlicher Forderungskatalog	5 Jahre Sicherheitsupdates für Smartphones, Auslieferung von Sicherheitspatches innerhalb eines Monats nach Release, Aussagepflicht zur Versorgung mit Betriebssystem-Updates, Ausstattung von Neugeräten mit neuestem Betriebssystem
Blauer Engel für Ressourcen- und energieeffiziente Softwareprodukte (2020)	Blauer Engel (2020) DE-UZ 215	Freiwilliges Umweltlabel	Explizite Anforderungen an potenzielle Hardware-Nutzungsdauer und Vorgaben für bedarfsgerechte Steuerung von Software: Verbot von Performance- und Hardwaredegradation durch Software, Software-Updates dürfen kein Hardware-Update bewirken, Gewährleistung von Abwärtskompatibilität, 5 Jahre verpflichtende Sicherheitsupdates nach Bereitstellungsende.
Warenkaufs-Richtlinie (EU 2019/771)	Richtlinie EU 2019/771 (2019)	Verpflichtende Richtlinie	Informationspflichten bei Sicherheitsaktualisierungen: „Im Falle von Waren mit digitalen Elementen sorgt der Verkäufer dafür, dass der Verbraucher über

Bezeichnung	Quelle	Policy-Format	Bezüge zu Softwareobsoleszenz
			<p>Aktualisierungen, einschließlich Sicherheitsaktualisierungen, die für den Erhalt der Vertragsmäßigkeit dieser Waren erforderlich sind, informiert wird und solche erhält.“ (Artikel 7 Absatz 3 RL EU 2019/771)</p> <p>Objekte Anforderungen an die Beschaffenheit der Ware: Waren müssen u.a. in ihrer Qualität, einschließlich Haltbarkeit, dem entsprechen was bei Waren der gleichen Art unter Berücksichtigung von Herstelleraussagen vernünftigerweise vom Verbraucher erwartet werden kann (Vgl. Artikel 7 Absatz 1 Buchstabe d RL EU 2019/771)</p>
Digitale-Inhalte-Richtlinie (EU 2019/770)	Richtlinie EU 2019/770 (2019)	Verpflichtende Richtlinie	<p>Subjektive und objektive Anforderungen an die Vertragsgestaltung und Beschaffenheit von digitalen Inhalten und Dienstleistungen (Art. 7 & Art. 8 RL EU 2019/770). Beinhaltet u.a. Angaben zu vereinbarten Merkmalen hinsichtlich Beschreibung, Qualität, Funktionalität, Kompatibilität, angestrebten Verwendungszweck sowie Bereitstellung von Anleitungen und Zubehör. Informations- und Bereitstellungspflicht von Aktualisierungen (Updatepflicht), einschließlich Sicherheitsaktualisierungen für den vertraglich geregelten Zeitraum und darüber hinaus im Rahmen des Gewährleistungszeitraums sowie was bei Waren der gleichen Art unter Berücksichtigung von Herstelleraussagen vernünftigerweise vom Verbraucher erwartet werden kann (Art. 8 Abs. 2 RL EU 2019/770).</p>
Ökodesign-Richtlinie (EU 2009/125)	RL (EU) 2009/125/EG	Verpflichtende Richtlinie	<p>Einzelne Produkt-Verordnungen enthalten spezifische Vorgaben für die Mindestverfügbarkeit von Firmware und Sicherheitsupdates von mindestens 7 Jahren und länger. z.T. sind auch Anforderungen für den Zugang zu Software, Reparatur und software-</p>

Bezeichnung	Quelle	Policy-Format	Bezüge zu Softwareobsoleszenz
Strategien gegen Obsoleszenz UBA Positionspapier 2017	Oehme et al. (2017)	Positionspapier mit Handlungsempfehlungen	relevanten Informationen enthalten. Herstellergarantienaussagepflicht zur Mindestlebensdauer von Produkten, Verbesserung der Interoperabilität von Software und Hardware zur Vermeidung funktionaler Obsoleszenz
Europäische Funkanlagenrichtlinie (RED)	RL (EU) 2014/53, radio equipment directive (RED)	Verpflichtende Richtlinie	Die Funkanlagenrichtlinien formuliert Anforderungen zur Beschaffenheit von Software und Informationen, die von Herstellern beim Inverkehrbringen von Funkanlagen (Smartphones, Radios, Router etc.) für einen bestimmungsgemäßen und EU-konformen Betrieb bereitgestellt werden müssen. Dies umfasst auch Softwareänderung und -updates nach der Inverkehrbringung der Produkte.
Leitfaden zur ISO 25010 Normenreihe Softwarequalität	ISO/IEC 25000:2014	Standard	Internationale Normenreihe zur Qualitätsbewertung von Softwareprodukten und Softwareentwicklungsprozessen.
IEEE-Standard 1061 - Softwarequalität	IEEE (1998)	Standard	Prozessstandard zur Formulierung von Qualitätskennzahlen für die formale Bewertung von Software

8.2.1 Bisherige umweltpolitische Leitlinien und Handlungsempfehlungen

Eine Schlüsselrolle und wesentlicher Akteur beim politischen Agenda-Setting spielte in Deutschland bisher das Umweltbundesamt, das bereits seit 2014 im Rahmen des Green-IT Projekts das Problem softwarebedingter Obsoleszenz als umweltpolitische Herausforderung benennen konnte (Gröger& Köhn 2015). Eine Studie zu Grüner Software im drauffolgenden Jahr wies zudem auf die softwarebedingte Obsoleszenz von Hardware hin, konnte jedoch nicht umfangreicher über Ursachen und Wirkungen aufklären (Hilty et al 2015). Die im Jahr 2016 veröffentlichte Obsoleszenzstudie des Umweltbundesamts stellte softwarebedingte Gründe als eine maßgebliche Spielart der funktionalen Obsoleszenz vor (Prakash et al. 2016: 146) und führte zu einem Positionspapier von Seiten des Umweltbundesamts, das unter anderem die Verbesserung der Interoperabilität von Hard- und Software für die Vermeidung funktionaler Obsoleszenz fordert (Oehme et al. 2017). Der Mitte 2020 vom Umweltbundesamt veröffentlichte Abschlussbericht zur „Weiterentwicklung von Strategien gegen Obsoleszenz einschließlich rechtlicher Instrumente“ stellt eine umfangreiche Bestandsanalyse zur rechtlich- und regulatorischen Situation vor, worin vereinzelt auch direkte Bezüge zur Softwareobsoleszenz zu finden sind (Keimeyer et al. 2020). Insbesondere wird auf bevorstehende Änderungen in der Ökodesign-Richtlinie hingewiesen, die für einige Produktgruppen der Zugang zu Test- und Fehlerdiagnosesoftware im Rahmen von Reparaturen ab dem Jahr 2021 erleichtern werden.

Zum Zeitpunkt der Erstellung der vorliegenden Studie sind diese Änderungen bereits in Kraft getreten. Weiterführend finden sich in der Literatur vereinzelt konkrete Vorgaben zum Verfügbarkeitszeitraum für Soft- und Firmwareupdates sowie Sicherheitsaktualisierungen (Keimeyer et al. 2019: S. 90ff.). Die Ökodesign-Anforderungen bei der Überarbeitung und Neuaufnahme einzelner Produktgruppen im Rahmen der Ökodesign-Richtlinie zeigen, dass das Thema zwischenzeitlich auch auf der EU-Ebene adressiert wird. Dies wird auch von der EU-Kommission vorangetrieben, die im aktuellen „New Circular Economy Action Plan“ neben einem Recht auf Reparatur, auch ein: „right to update obsolete software“ (EC 2020b) anstrebt.

Anfang 2020 stellte das Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit (BMU) seine umweltpolitische Digitalagenda vor. Die Agenda definiert strategische Grundsätze und Ziele, um die Digitalisierung nachhaltig und sozialökologisch zu gestalten. Sie enthält darüber hinaus einen umfangreichen Maßnahmenkatalog, in dem stellenweise auch auf das Problem softwarebedingter Obsoleszenz eingegangen wird. Hierunter zählen u.a. die Forderung nach verbindlichen Herstellervorgaben in der EU-Ökodesign-Richtlinie, um eine möglichst lange Funktionsfähigkeit von Hardware-Software-Systemen zu gewährleisten; verbindliche Regelungen zur Bereitstellung von Updates als Teil einer Garantieaussagepflicht und eines Rechts auf Reparatur sowie die Förderung von „Grünem Coding“ und Methoden der ressourcenschonenden Softwareentwicklung (BMU 2020).

Erwähnenswert, wenn auch nicht im direkten Zusammenhang mit den umweltpolitischen Zielen des BMU und UBA, ist die Veröffentlichung eines Forderungspapier zur Sicherheit von Smartphones durch das Bundesamt für Sicherheit in der Informationstechnik (BSI) im Februar 2020, in dem unter anderem 5 Jahre Sicherheitsupdates, die Auslieferung von Sicherheitspatches innerhalb eines Monats nach Release, eine Aussagepflicht zur Versorgung mit Updates des Betriebssystems und die Ausstattung von Neugeräten mit neuestem Betriebssystem gefordert wird (BSI 2020a). Der Anforderungskatalog richtet sich als Diskussionsgrundlage an Hersteller, Erstausrüster, Netzbetreiber und Zivilgesellschaft. Die Anforderungen sollen nach Wunsch des BSI unter Beteiligung aller gesellschaftlichen Gruppen weiterentwickelt werden und die zukünftige Richtlinie für die Erteilung des von der Bundesregierung geplanten IT-Sicherheitskennzeichens für Smartphones (BSI 2020b).

Auffällig ist vor diesem Hintergrund, dass eine verpflichtende Versorgung mit Softwareupdates und die Gewährleistung der Funktionsfähigkeit von Hardware-Software-Systemen über einen bestimmten Mindestzeitraum politisch nicht einzig aus umwelt- und verbraucherrechtlichen Aspekten, sondern auch auf Basis von sicherheitsrelevanten Fragestellungen adressiert wird. Letzteres spielt insbesondere im industriellen Umfeld und bei Anlagenbetreibern aufgrund der gestiegenen Softwareabhängigkeit eine zunehmende Rolle (Bartels et al. 2019).

8.2.2 Warenkaufrichtlinie (EU 2019/771) und Digitale-Inhalte-Richtlinie (EU 2019/770)

Der Verkauf von Waren (Produkte und Dienstleistungen) wird im rechtlichen Sinn durch Kaufverträge geregelt, woraus sich Pflichten für Käuferinnen und Verkäuferinnen ergeben. Hierzu zählt unter anderem auch die Beschaffenheit der Ware (Qualität) zum Übergabezeitpunkt (Verkauf) und stellt insbesondere für die Bewertung von vorzeitiger (Software-)Obsoleszenz eine Herausforderung dar (Hess 2018: 72ff.). Der Verbrauchsgüterkauf wird seit 2019 europaweit von der sogenannten Warenkauf-Richtlinie (Richtlinie (EU) 2019/771) einheitlich geregelt und muss von den Mitgliedsstaaten bis Juli 2021 in nationales Recht umgesetzt werden. Ergänzt wird die Richtlinie durch Digitale-Inhalte-Richtlinie (EU 2019/770), welche auch eigenständige („stand-alone“) digitale Inhalte erfasst, während die Warenkaufrichtlinie für digitale Inhalte gilt, die bereits Bestandteil eines beweglichen Produkts sind (z.B. Firmware, Betriebssysteme) oder mit diesem verbunden sind und im Rahmen desselben Vertrags verkauft

werden. Zusammengenommen ergeben sich auf Basis beider Richtlinien neue Informations- und Bereitstellungspflichten für Hersteller und Inverkehrbringer von software-relevanten Produkten.

Die Warenkauf-Richtlinie regelt neben den Gewährleistungsrechten beim Warenkauf neuerdings auch Anforderungen an die Bereitstellung von digitalen Inhalten, worunter auch Softwareanwendungen oder Teile davon zählen können, wenn sie bereits Gegenstand der Ware sind. Die Richtlinie formuliert unter anderem eine Informationspflicht zu Sicherheitsaktualisierungen: „Im Falle von Waren mit digitalen Elementen sorgt der Verkäufer dafür, dass der Verbraucher über Aktualisierungen, einschließlich Sicherheitsaktualisierungen, die für den Erhalt der Vertragsmäßigkeit dieser Waren erforderlich sind, informiert wird und solche erhält (Artikel 7 Absatz 3 RL EU 2019/771)“. Dies gilt zumindest für den gesetzlichen Gewährleistungszeitraum, der in Deutschland 2 Jahre beträgt und darüber hinaus für freiwillige Garantien durch den Garantiegeber. Zudem werden objektive Anforderungen an die Beschaffenheit der Ware gestellt, in dem Waren u.a. in ihrer Qualität, einschließlich Haltbarkeit, dem entsprechen müssen, was bei Waren der gleichen Art, unter Berücksichtigung von Herstelleraussagen vernünftigerweise vom Verbraucher*innen erwartet werden kann (Vgl. Artikel 7 Absatz 1 Buchstabe d RL EU 2019/771).

Gegenüber der Warenkaufrichtlinie, umfasst die Digitale-Inhalte-Richtlinie nicht nur digitale Inhalte und Leistungen, die bereits beim Erwerb eines beweglichen Produkts (z.B. Smartphone, Laptop) Gegenstand der Sache oder mit diesem verbunden sind, sondern auch digitale Inhalte, die unabhängig hiervon („stand-alone“) erworben werden können (z.B. Spiele, Apps, Software, Videos, Streaming Abos). Die Richtlinie stellt subjektive und objektive Anforderungen an die Vertragsmäßigkeit, wozu insbesondere Angaben zur erwartbaren Beschaffenheit hinsichtlich der Beschreibung, Qualität, Funktionalität oder Kompatibilität zählen, aber auch die Bereitstellung von Anleitungen und Zubehör. Analog zur Warenkaufrichtlinie sieht die Digitale-Inhalte-Richtlinie ebenfalls eine Updatepflicht vor, welche Anbieter digitaler Leistungen dazu verpflichtet, Verbraucher*innen „über Aktualisierungen einschließlich Sicherheitsaktualisierungen, die für den Erhalt der Vertragsmäßigkeit der digitalen Inhalte und digitalen Dienstleistungen erforderlich sind“, zu informieren und diese bereitzustellen (Art. 8 Abs. 2 RL EU 2019/770).

Es obliegt der jeweiligen nationalen Gesetzgebung und Rechtsprechung, welche Haltbarkeiten hier „vernünftigerweise“ von Verbraucher*innen erwartet werden können, hierzu zählt auch die erwartbare Softwarequalität. Eine Richtschnur und passenden rechtlichen Rahmen bietet hierfür die Ökodesign-Richtlinie ((EU) 2009/125/EG), in welcher europaweit bereits produktspezifische Anforderungen an energieverbrauchsrelevante Produkte gestellt werden können.

8.2.3 Ökodesign-Richtlinie (RL (EU) 2009/125/EG)

Die Ökodesign-Richtlinie ermöglicht produktgruppenspezifische Verordnungen mit Anforderungen für energieverbrauchsrelevante Produkte. In diesen werden u.a. einheitliche Vorgaben zur umweltgerechten Gestaltung gemacht werden. Die produktgruppenspezifischen Verordnungen werden regelmäßig überarbeitet. Für erste Produkte gelten bereits seit März 2021 Vorgaben zur Verfügbarkeit und dem Zugang zu Soft- und Firmware:

Tabelle 18: Anforderungen an software-relevante Aspekte in den Ökodesign-Richtlinien (RL (EU) 2009/125/EG)

Verordnung	Mindestzeitraum für Support [Jahre]*	Beschränkter Zugang**	Zusätzliche softwarebezogene Anforderungen und Informationen
Server und Datenspeicherprodukte ((EU) 2019/424)	8	Nein	Neuste Firmware wird kostenlos oder zu fairen, transparenten, nichtdiskriminierenden Kosten zur Verfügung gestellt Sicherheitsaktualisierungen werden kostenlos zur Verfügung gestellt
Elektronische Displays ((EU) 2019/2021)	8	Nein	Neuste Firmware wird kostenlos oder zu fairen, transparenten, nichtdiskriminierenden Kosten zur Verfügung gestellt Informationen über die garantierte Mindestverfügbarkeit von Software- und Firmware-Updates, die Verfügbarkeit von Ersatzteilen und die Produktunterstützung sind im Produktinformationsblatt anzugeben Nach einer Software- oder Firmware-Aktualisierung dürfen sich der Energieverbrauch des Produkts und alle anderen angegebenen Parameter nicht ohne vorige Zustimmung der Nutzer*in ändern
Haushaltsgeschirrspüler ((EU) 2019/2022)	7	Ja	Netzwerkverbindungen müssen deaktiviert werden können; Betrieb auch ohne Netzwerkverbindung; Zugang zu Software als Ersatzteil kann auf gewerbliche Reparatur beschränkt werden; Fachlich kompetente Reparateure erhalten innerhalb eines Werktages Zugang zu reparaturrelevanten Informationen
Haushaltswaschmaschinen und Haushaltswaschtrockner (RL (EU) 2019/2023)	10	Ja	Netzwerkverbindungen müssen deaktiviert werden können; Betrieb auch ohne Netzwerkverbindung; Zugang zu Software als Ersatzteil kann auf gewerbliche Reparatur beschränkt werden; Gewerbliche Reparateur erhalten innerhalb eines Werktages Zugang zu reparaturrelevanten Informationen

Verordnung	Mindestzeitraum für Support [Jahre]*	Beschränkter Zugang**	Zusätzliche softwarebezogene Anforderungen und Informationen
Kühlgeräte mit Direktverkaufsfunktion ((EU) 2019/2024)	8	Ja	Fachlich kompetente Reparateure erhalten innerhalb eines Werktages Zugang zu reparaturrelevanten Informationen
Schweißgeräte ((EU) 2019/1784)	10***	Ja	Fachlich kompetente Reparateure erhalten innerhalb eines Werktages Zugang zu reparaturrelevanten Informationen

* Mindestzeitraum nach dem Inverkehrbringen des letzten Exemplars eines Modells.

** Zugang zu Reparatur- Wartungsinformationen kann zwei Jahre nach dem Inverkehrbringen des letzten Exemplars eines Modells auf fachlich kompetente Reparateure beschränkt werden.

*** Nach der Herstellung des letzten Exemplars eines Schweißgerätemodells.

Um die Produktgruppen übergreifenden Aspekte im Ökodesign noch besser zu adressieren, ließ die Europäische Kommission im Rahmen des Arbeitsplans für Ökodesign und Energieverbrauchskennzeichnung 2020-2024 eine Vorbereitungsstudie durchführen, aus der Empfehlungen für eine horizontale Anforderung für die Mindestverfügbarkeit von Soft- und Firmware hervorgehen.¹⁷

8.2.4 Europäische Funkanlagenrichtlinie (RED) RL (EU) 2014/53

Die europäische Funkanlagenrichtlinie (Radio Equipment Directive - RED) formuliert u.a. Anforderungen zur Beschaffenheit von Software und Informationen, die von Herstellern beim Inverkehrbringen von Funkanlagen für einen bestimmungsgemäßen und EU-konformen Betrieb bereitgestellt werden müssen. Dies umfasst auch Softwareänderung und -updates nach der Inverkehrbringung der Produkte in den europäischen Markt. Weil ein Großteil der vernetzten Geräte heute in irgendeiner Form Funkstandards und -technologien nutzt (Wlan, Bluetooth, NFC, Mobilfunk, GPS) bezieht sich die Norm auf eine Vielzahl an vernetzten Geräten, wie Smartphones- und Mobiltelefone, Router, Smartwatches, Tablets oder Radios. Aufgrund der steigenden Bedeutung von Software in Funkanlagen und die Möglichkeit, diese durch Software im Nachhinein zu rekonfigurieren, veröffentlichte die EU-Kommission im Zuge einer Initiative im Jahr 2020 ein Begleitdokument und Leitfaden, der sich dezidiert mit der Regulierung von Software in der Richtlinie auseinandersetzt und auf verschiedene Fragen eingeht (EC 2020a). Das Begleitdokument nimmt keinen expliziten Bezug auf Softwareobsoleszenz vor, es können jedoch folgende Aspekte herausgestellt werden:

- ▶ Hersteller von Funkanlagen und Software sind verpflichtet spezifische Informationen zum bestimmungsgemäßen Betrieb bereitzustellen. Hierzu zählen insbesondere technische Unterlagen, die mindestens Angaben zur Software- und Firmwareversionen enthalten.
- ▶ Softwareänderungen und -updates können bestimmte Aspekte der Funkanlage steuern und so verändern, dass kein sicherer Betrieb oder Zugang zu Notdiensten möglich ist. Bei der

¹⁷ EC (European Commission) (2021): Preparatory study for the Ecodesign and Energy Labelling Working Plan 2020-2024. Task 4 Complementary analyses and recommendations, Download: <https://drive.google.com/file/d/1pL20BbMkaYBEbOJ5HUgsqsVQHIncxoJP/view>

Entwicklung und Inverkehrbringung ist daher auf eine flexible Hardware zu achten, die je nach hochgeladener Software unterschiedliche Konfigurationen annehmen kann.

- Die Konformität der Funkanlage soll auch dann gewährleistet bleiben, wenn Nutzer*innen oder Dritte neue Software hochladen und damit die Konfiguration des Geräts ändern können.

Inwiefern heute bereits in der Praxis Gefahren und Konformitätsverletzungen aufgrund der Rekonfiguration von Funkanlagen durch Softwareänderungen resultieren ist empirisch bisher wenig erforscht. Das Begleitdokument nennt jedoch Szenarien, in denen beispielweise durch Änderungen der Software höhere Leistungsaufnahmen auf Hardwareebene realisiert werden können, die entweder andere Frequenzen stören und/oder die Sendeleistung verändern können. Expert*innen warnen bereits, dass mit der Zunahme an vernetzten Geräten und dem damit verbundenen Funkverkehr, dem sogenannten „Internet of Things (IoT)“, elektromagnetische Störungen (Electromagnetic interference - EMI) zunehmen, welche Geräte in ihrer Funktionsfähigkeit beeinträchtigen können. Probleme resultieren dabei auf beiden Seiten: Sender („dirty transmitters“) können bei falscher Konfiguration unmittelbar Störungen bei anderen Geräten auslösen. Gleichzeitig sind unzureichend konfigurierte Empfänger besonders anfällig für EMI, die mit einer Zunahme der „Inference of Things“ jedoch unvermeidbar sind (Rowe 2017). Die Fähigkeit durch Software, das Sende- und Empfängerverhalten von Geräten im Nachhinein zu verändern, birgt somit Risiken und Potenziale zur Vermeidung von Obsoleszenz gleichermaßen.

8.2.5 Relevante Initiativen und Gesetze in anderen Ländern

In Frankreich wurde im Jahr 2015 ein Gesetz verabschiedet, in dem geplante Obsoleszenz unter Strafe gestellt wird (LOI n° 2015-992). Obwohl bisher nur wenige Hersteller in der Praxis mit einer Klage konfrontiert wurden, gab es bereits Ermittlungen gegen den Hersteller Apple, was in der Folge zur Zahlung eines Bußgelds in Höhe von 25 Millionen Euro geführt hat und in Zusammenhang mit der softwarebasierten Leistungsdrosselung von iPhones steht (DG CCRF 2020). „Nachdem Nutzer sich Ende 2017 nach einem Softwareupdate über den Verdacht der Softwaredrosselung bei älteren iPhone-6-Smartphones beschwert haben, erklärte der Hersteller Apple, dass es sich in diesem Fall um eine Schutzmaßnahme handele, da die bei älteren Akkus unvermeidbaren Kapazitätsverluste bei sonst gleicher Nutzungsintensität zum vorzeitigen Abschalten der Geräte führen könnten“ (Poppe 2019: 26). Die für die Überprüfung zuständige Regierungsbehörde warf dem Hersteller insbesondere irreführende Geschäftspraktiken vor und die Nutzenden nicht ausreichend über die Folgen der Softwaredrosselungen informiert zu haben (DG CCRF 2020). Eine vergleichbare Verurteilung gab es im Vorfeld hierzu bereits 2018 in Italien gegenüber Apple und Samsung (AGCM 2018). In den USA stimmte Apple Anfang 2020 einem Vergleich von 500 Millionen US-Dollar zu (Steinberg 2020).

Die Urteile haben gemeinsam, dass die Hersteller nicht für die softwarebedingte Drosselung der Geräte verurteilt wurden, sondern primär für die unzureichende Informierung der betroffenen Nutzenden. Softwarebasiertes Batteriemangement wird von vielen Herstellern sogar explizit beworben und soll in der Regel die mögliche Lebensdauer der Geräte erhöhen (Poppe & Longmuss 2019: 26). Das Beispiel macht jedenfalls deutlich, dass Aussage- und Informationspflichten zum Einfluss von Software auf die Gesamtperformance durchaus ihre Berechtigung haben – nicht nur um vorzeitige Obsoleszenz zu vermeiden, sondern vor allen, um Rechtssicherheit für Hersteller und Verbraucher*innen zu schaffen.

Eine Möglichkeit die Rechtssicherheit in diesem Bereich zu stärken, bietet beispielsweise der ab dem Jahr 2021 verpflichtende Reparatur-Index in Frankreich, welcher Verbraucher*innen

Auskunft darüber geben soll, welche Möglichkeiten zur Reparatur im Schadensfall möglich sind. Der Reparatur-Index nimmt eine Bewertung verschiedener Kriterien vor und ermittelt daraus einen Reparatur-Score, der sichtbar für Verbraucher*innen am Verkaufsprodukt angebracht werden muss. In die Bewertung fließt dabei als Teilkriterium auch die Verfügbarkeit von Software-Updates und Möglichkeit für Software-Resets ein. Der Score wird von den Herstellern selbst berechnet (Heinz & Meyer 2020).

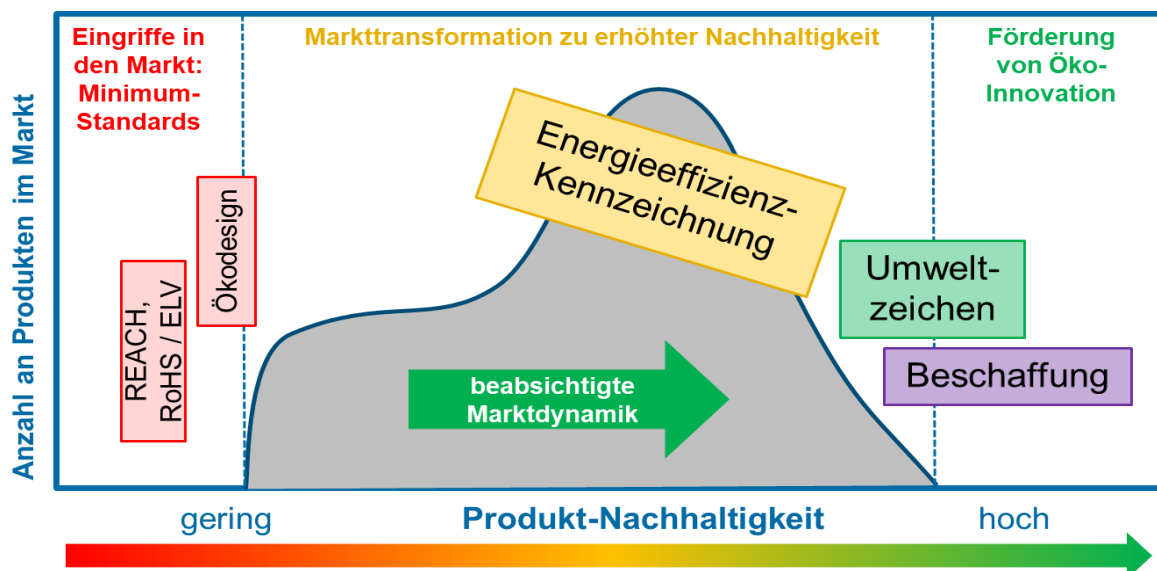
Eine weiterführende Betrachtung sollte insbesondere untersuchen, ob Verbraucher*innen sich an Produktkennzeichnungen zur Nutzungsdauer bei ihrer Kaufentscheidung orientieren und welche Informationen und Aussagen von Herstellern bereits im Zuge von Softwareupdates bereits gemacht werden und mit welchen Anforderungen diese ggf. sinnvoll in Zukunft ergänzt werden können.

8.3 Umweltpolitische Handlungsempfehlungen

Die Verkürzung der Lebensdauer von Produkten durch softwarebedingte Obsoleszenz ist sowohl ein Thema des Ressourcenschutzes als auch ein Thema des Verbraucherschutzes. Daher sind grundsätzlich alle produktpolitischen Instrumente, die diese beiden Themen adressieren, geeignet, das Problem der softwarebedingten Obsoleszenz einzudämmen.

In Abbildung 30 werden verschiedene Ansatzpunkte von Politikinstrumenten zur nachhaltigen Markttransformation dargestellt. Die Abbildung stellt die Funktion einer theoretische Marktverteilung von Produkten entlang einer Nachhaltigkeits-Skala dar, wobei unterstellt wird, dass es sehr viele „mittelmäßig“ nachhaltige Produkte gibt und jeweils keine Produkte, die eine gewisse Produkt-Nachhaltigkeit unter- oder überschreiten. Die eigentliche Verteilung des Marktes zwischen diesen beiden Extrempunkten (minimal und maximal) ist in der Abbildung rein hypothetisch. In der Abbildung 30 wird unterstellt, dass es im Wesentlichen zwei Marktsegmente gibt, Produkte mit niedrigem Nachhaltigkeitsstandard (linker Teil der Kurve) und solche mit höheren Nachhaltigkeitseigenschaften (rechts der Mitte). Die Zahl der Produkte mit überdurchschnittlichen Nachhaltigkeitseigenschaften nimmt nach rechts kontinuierlich ab, um ganz rechts beim Maximum der Nachhaltigkeitseigenschaften eine Minimalzahl an Produkten zu erreichen.

Abbildung 30: Politikinstrumente zur nachhaltigen Markttransformation



Quelle: Eigene Darstellung, Öko-Institut e.V. nach einer Grafik von DG Environment

Bei den Politikinstrumenten kann unterschieden werden, in solche, die in den Markt eingreifen, indem sie Minimumstandards setzen und unerwünschte Produkte vom Markt ausschließen (links in der Abbildung 30). Solche Instrumente sind beispielsweise die Verordnungen und Richtlinien auf europäischer Ebene (z.B. REACH, RoHS, ELV, Ökodesign-VOen)¹⁸, die erfüllt sein müssen, damit ein Produkt mit dem CE-Konformitäts-Kennzeichen („Conformité Européenne“) gekennzeichnet und auf den europäischen Markt gebracht werden darf.

Andere Politikinstrumente sind solche, die den Wettbewerb fördern. Innerhalb der am Markt angebotenen Produkte (Mitte der Abbildung 30) können vor allem Transparenzmaßnahmen dabei unterstützen, dass sich der Markt in eine nachhaltigere Richtung bewegt. Die Energieeffizienz-Kennzeichnung ist ein solches Instrument, die Produkte hinsichtlich ihrer Energieeffizienz unterscheidbar macht und damit einen Wettbewerb um die energieeffizientesten Produkte fördert. Die Markttransformation kann aber auch mit fiskalischen oder finanziellen Instrumenten erreicht werden, beispielsweise durch Energiesteuern, Pflichtpfand für Produkte oder durch die Einführung eines Handels für Verschmutzungsrechte. Diese beispielhaft genannten Instrumente würden den Energieverbrauch senken, das Entsorgen verhindern und somit eine Nachnutzung oder ein Recycling ermöglichen sowie Umweltverschmutzung reduzieren. Die umweltverträgliche öffentliche Beschaffung kann durch den gezielten Einkauf umweltverträglicherer Produkte Anreize dafür schaffen, solche Produkte vermehrt anzubieten.

Der dritte Ansatzpunkt für Produktpolitik ist die Förderung von Öko-Innovationen (rechts in der Abbildung 30), also die Schaffung von besseren Voraussetzungen für besonders nachhaltige Produkte. Erreicht wird dies beispielsweise durch Umweltzeichen oder die Förderung von Produkten, die wegen fehlender Wirtschaftlichkeit, noch nicht am Markt verfügbar sind. Die umweltverträgliche öffentliche Beschaffung bietet beispielsweise die Möglichkeit, besonders umweltverträgliche Produkte pilothaft zu fördern und ihnen damit einen Marktzutritt zu ermöglichen.

Diese drei Ansatzpunkte werden in den drei nachfolgenden Kapiteln in konkrete produktpolitische Maßnahmen bezogen auf softwarebedingte Obsoleszenz übersetzt.

8.3.1 Mindestanforderungen für den Marktzugang

Zur verbindlichen Festlegung von Mindestanforderungen an Produkte, die von Software abhängig sind, bieten sich die Ökodesign-Richtlinie (jeweils bezogen auf einzelne Produktgruppen), aber auch die anderen in Tabelle 18 genannten Richtlinien und Verordnungen an, die sich auf eine Vielzahl von Produkten beziehen (z.B. Warenkauf-Richtlinie, Funkanlagen-Richtlinie und Digitale-Inhalte-Richtlinie). Ebenso ist es denkbar, die Themen in einer neu zu schaffenden Verordnung zur Verbesserung des Ressourcenschutzes (bzw. Materialeffizienz) unterzubringen und damit zur Umsetzung des Circular Economy Action Plans beizutragen.

Folgende Themen sollten bei Ausgestaltung von Verordnungen aus Sicht der Autor*innen der vorliegenden Studie adressiert werden:

¹⁸ REACH: Registration, Evaluation, Authorisation and Restriction of Chemicals; RoHS: Restriction of Hazardous Substances in Electronic Equipment; ELV: End of Life Vehicles Directive

1. Möglichkeit zum Betrieb der Geräte ohne externe Abhängigkeiten

- ▶ Softwarebetriebene Geräte sollten grundsätzlich ohne externe Abhängigkeiten betrieben werden können.
- ▶ Dies beinhaltet, dass die Geräte ihre Kernfunktionalität ohne Netzwerkanbindung („offline“) und ohne externe Cloud-Dienste erbringen müssen.
- ▶ Der volle Funktionsumfang darf durch die Entkopplung von externen Abhängigkeiten eingeschränkt sein (Fallback-Option).
- ▶ Die Hersteller müssen jedoch beim Verkauf der Produkte die Kernfunktionalitäten benennen, also welchen Funktionsumfang die Geräte im entkoppelten, unabhängigen Betriebsmodus noch erbringen.

2. Bereitstellung von sicherheitsrelevanten Software-Updates über einen Mindestzeitraum

- ▶ Hersteller von Produkten sollten dazu verpflichtet werden, über einen Mindestzeitraum Software-Updates bereit zu stellen, die einen sicheren Betrieb der Produkte zulassen (Schutz der Gesundheit, der Privatsphäre und vor finanziellem Verlust).
- ▶ Der Zeitraum beginnt mit dem Ende des Inverkehrbringens des letzten Exemplars eines Modells der jeweiligen Produkte und sollte mindestens 10 Jahre betragen.

Einen Schritt weiter als der im vorangehenden Punkt genannte Mindestzeitraum für Software-Updates geht die Forderung nach einer gesetzlichen Mindestnutzungsdauer.

3. Gewährleistung einer Mindestnutzungsdauer

- ▶ Der Hersteller von softwarebetriebenen Produkten muss gewährleisten, dass seine Produkte über einen Mindestzeitraum in gleichbleibender Qualität nutzbar sind und muss die entsprechenden Voraussetzungen dafür schaffen.
- ▶ Innerhalb der Mindestnutzungsdauer besteht für den Hersteller die Pflicht zu Software-Updates für den Erhalt aller Funktionen, die zum Übergabezeitpunkt zur Verfügung standen. Software-Updates dürfen die Ausführungsgeschwindigkeit der Funktionen des Produktes nicht mehr als 30% reduzieren und den Energiebedarf insgesamt nicht mehr als 10% erhöhen.
- ▶ Zur Gewährleistung der Mindestnutzungsdauer gehören insbesondere die vertraglichen Vereinbarungen mit eigenen Lieferanten und externen Dienstleistern und mit Herstellern von Teilprodukten eines zusammenhängenden Produktsystems. Dies sind beispielsweise die Ersatzteilbereitstellung (z.B. Akkus, Speichermedien, abnutzende Bauteile, Steckernetzteile), Bereitstellung kompatibler Produkte eines Produktsystems (z.B. Datenbrücken/Bridges, zentrale Steuereinheiten, verteilte Sensoren und Aktoren oder Fernbedienungen), Bereitstellung von Software-Updates (z.B. für die Bedienbarkeit des Produkts selbst und die sichere Datenübertragung), Aufrechterhaltung externer Cloud-Dienste (z.B. zur Datenlieferung, -verarbeitung und -speicherung). Vertragspartnerschaften zusammengehöriger Dienste und Geräte von Produktsystemen müssen entsprechend ausgestaltet werden.
- ▶ Diese Mindestnutzungsdauer sollte deutlich über die gesetzliche Gewährleistungsfrist von 2 Jahren hinaus gehen und sich an der technischen Lebensdauer der Hardware orientieren. Auch

bei Geräten, die unerwartet lange physisch einwandfrei sind, sollte Software nicht zur Obsoleszenz führen.

- ▶ Es wird eine Mindestnutzungsdauer von regelmäßig 10 Jahren vorgeschlagen, die nur in begründeten Ausnahmefällen kürzer sein darf.

4. Verbot von softwareseitigem Verhindern von Reparatur und Betrieb

- ▶ Die in Produkten integrierte Software darf nicht dazu genutzt werden, die Reparatur und den Betrieb von Produkten zu verhindern, indem beispielsweise Ersatzteile oder Betriebsmittel anderer Hersteller ausgeschlossen werden („parts pairing“).
- ▶ Sofern die Software für die Betriebssicherheit der Geräte zuständig ist (z.B. Überladungsschutz für Akkus) müssen die Standards offengelegt werden, die Reparaturbetriebe und Ersatzteilhersteller in die Lage versetzen, ebenso sichere Produkte einzubauen.
- ▶ Die Software darf ebenfalls nicht dazu genutzt werden, die Nutzungsdauer von Produkten zu begrenzen (z.B. Countdown- oder Abschaltvorrichtungen) oder deren Qualität und Funktionalität zu reduzieren.

5. Anforderungen an die Kompatibilität und Interoperabilität von Produktsystemen

- ▶ Der Hersteller von Produkten, die zu einem Produktsystem gehören (z.B. einzelne Steuereinheiten, Sensoren und Aktoren einer Smart-Home-Anwendung), müssen dafür Sorge tragen, dass ihr Produkt über die Mindestnutzungsdauer mit dem zugehörigen Produktsystem zusammen genutzt werden kann, auch dann, wenn sich das Produktsystems weiterentwickelt.
- ▶ Dies muss er entweder durch die Aufrechterhaltung der Kompatibilität seines Produktes zu dem zugehörigen Produktsystem (beispielsweise durch Software-Updates am Produkt selbst) oder durch vertragliche Vereinbarungen mit dem Hersteller des Produktsystems erreichen (z.B. durch Vereinbarung der Rückwärtskompatibilität des Produktsystems zu früheren Produkten).
- ▶ Die Pflicht zu vertraglichen Vereinbarungen mit Dritten entfällt, wenn im Produkt allgemein anerkannte und offen zugängliche Standards verwendet werden, die auch mit anderen Produktsystemen kompatibel sind (z.B. genormte physische Schnittstellen und offene Übertragungsprotokolle).
- ▶ Die Hersteller von Produktsystemen müssen ihrerseits zur Kompatibilität beitragen, indem sie anderen Marktteilnehmern die Interoperabilität mit ihrem Produktsystem ermöglichen, z.B. indem die Schnittstellen und Übertragungsprotokolle des Produktsystems offenen Standards entsprechen und öffentlich zugänglich dokumentiert werden.

6. Anforderungen an externe Dienstleistungen

- ▶ Sind zum Betrieb eines softwarebetriebenen Produktes externe Dienstleistungen erforderlich, beispielsweise eine externe Cloud-Dienstleistung, so muss der Hersteller eines derart abhängigen Produktes durch vertragliche Vereinbarungen mit dem externen Dienstleister sicherstellen, dass die externe Dienstleistung für mindestens 10 Jahre aufrechterhalten wird.
- ▶ Diese Anforderung kann entfallen, wenn das Produkt auch ohne externe Abhängigkeiten in gleichbleibender Qualität betrieben werden kann.

7. Anforderungen an Verkaufs- und Vertriebspattformen von Software/Apps

- ▶ Verkaufs- und Vertriebspattformen von Software/Apps dürfen ihre marktbeherrschende Stellung nicht dazu missbrauchen, softwarebedingte Obsoleszenz herbei zu führen.
- ▶ So sollten Verkaufs- und Vertriebspattformen, die eine vorherrschende Marktstellung aufweisen (z.B. Marktanteil von mehr als 50% für bestimmte Produkte), durch ihre Vertragsbedingungen keine Apps von dem Vertrieb über ihre Plattform ausschließen, die für ältere Hardware und für ältere Betriebssysteme programmiert sind. Dadurch soll sichergestellt werden, dass auch ältere Geräte mit neuer Software bzw. Software-Updates versorgt werden können.
- ▶ Entwicklern von open-source Software muss es kostenfrei gestattet und ermöglicht werden, nicht-kommerzielle Software und Apps für alle Betriebssysteme und Versionen zur Verfügung zu stellen, sodass diese Software von Konsument*innen kostenfrei installiert und genutzt werden kann.

8.3.2 Transparenzmaßnahmen zur Förderung von Wettbewerb

Transparenzmaßnahmen tragen dazu bei, dass Produkte am Markt unterscheidbar und ihre (Umwelt-) Eigenschaften quantifizierbar werden. Sowohl private Konsument*innen als auch professionelle Einkäufer*innen (Unternehmen, öffentliche Hand) werden dadurch in die Lage versetzt, solchen Produkten den Vorzug zu geben, die beispielsweise besonders energieeffizient, langlebig und ressourcenschonend sind. Die Transparenzmaßnahmen sollten dabei nicht freiwillig sein, sondern verbindlich, so wie dies im Bereich der energierelevanten Produkte (Energieeffizienz-Kennzeichnung), der Gebäude (Energieausweis) aber auch im Bereich der Lebensmittel (Deklaration der Inhaltsstoffe) marktüblich ist.

Die Autor*innen der vorliegenden Studie schlagen vor, folgende Transparenzmaßnahmen verbindlich einzuführen:

8. Nennung der Abhängigkeiten von softwarebetriebenen Produkten

- ▶ Es sollte durch den Hersteller oder Händler des jeweiligen Produktes Transparenz darüber geschaffen werden, welche zusätzliche Hardware, Software oder Dienstleistung für den vollen Funktionsumfang des Produktes erforderlich sind (z.B. Notwendigkeit eines Smartphones unter Angabe des Betriebssystems und dessen Version, erforderliche Bandbreite im Netzwerk).
- ▶ Pflicht zur Nennung der erforderlichen externen Dienstleistungen oder Cloud-Services, die zur Erfüllung des vollen Funktionsumfangs benötigt werden (z.B. Benutzerkonto bei einem Cloud-Service).
- ▶ Pflicht zur Nennung von Art und Umfang der über Netzwerke versendeten Daten, Zweck der externen Verarbeitung und Angaben zum Datenschutz.

9. Verpflichtende Angabe des garantierten Supportzeitraums und daraus resultierende Rechtsansprüche

- ▶ Hersteller von softwarebetriebenen Produkten sollten beim Verkauf von Produkten und vernetzten Produktsystemen verpflichtet werden, den garantierten Supportzeitraum für Software-Updates durch die Nennung des Support-Enddatums anzugeben und zu beschreiben, worauf sich dieser Support bezieht.

- ▶ Während des genannten Zeitraums vom Kauf bis zum Support-Enddatum haben die Käufer*innen des Produktes einen Rechtsanspruch auf die Bereitstellung von Software-Updates und der genannten Unterstützungen. Dieser Anspruch gilt auch für gebraucht gekaufte Geräte.
- ▶ Die Verpflichtung zur Deklaration des garantierten Supportzeitraums kann bereits eingeführt werden, bevor es einen Mindestzeitraum für den Software-Support oder eine Mindestnutzungsdauer gibt (siehe Punkt 3: Gewährleistung einer Mindestnutzungsdauer).
- ▶ Für den Fall, dass der Hersteller keinen Support über die gesetzliche Gewährleistungsfrist hinaus anbietet, muss dies ebenfalls deklariert werden (z.B. „kein Support“).

10. Kennzeichnung von Produkten zu ihrer Reparierbarkeit

- ▶ Softwarebetriebene Produkte sollen europaweit mit einer leicht verständlichen Kennzeichnung über ihre Reparierbarkeit versehen werden, nach der Art des französischen Reparatur-Scores (vgl. Kapitel 8.2.5).
- ▶ Die garantierte Supportdauer und die Wartbarkeit der im Produkt enthaltenen Software sollten zusätzlich im Reparatur-Score berücksichtigt werden. Wartbare Software kann z.B. ohne Einschränkungen sowohl up-graded (in eine höhere Version versetzt) als auch down-graded (in eine niedrigere Version versetzt) werden und ist modular aufgebaut.

11. Nennung der übertragenen Datenmengen und Art der Übertragungsinhalte

- ▶ Softwarebetriebene Produkte, die mit einem Netzwerk verbunden sind, müssen die übertragene Datenmenge für ein typisches, vom Hersteller vorgegebenes Nutzungsszenario (Datenmenge pro Jahr) oder für den regelmäßigen Betriebszustand des Geräts (Übertragungsrate) angeben.
- ▶ Es muss eine Angabe erfolgen, welche Daten zu welchem Zweck übertragen werden (z.B. Temperaturen, Audioaufzeichnungen, Bilder, Bewegungsdaten, personenbezogene Daten usw.) und wie die Daten geschützt bzw. verwertet werden.

8.3.3 Förderung von Öko-Innovationen

Die Förderung von besonders ressourcenschonenden Produkten bietet die Möglichkeit, die Innovationskraft von Unternehmen zu unterstützen und den Markt insgesamt in Richtung nachhaltigerer Produkte zu bewegen. Klassische Instrumente sind hierfür Umweltzeichen (z.B. Blauer Engel), Plattformen für herausragende Produkte (z.B. ecotopten.de) und die umweltverträgliche öffentliche Beschaffung aber auch Subventionen, Auszeichnungen, Steuererleichterungen und Förderung.

Die Autor*innen dieser Studie schlagen folgende Maßnahmen vor, um besonders langlebige und ressourcenschonende softwarebetriebene Produkte zu fördern:

12. Verpflichtende umweltverträgliche öffentliche Beschaffung

- ▶ Die öffentliche Beschaffung von umweltverträglichen Produkten ist derzeit weitestgehend optional oder steht in Konkurrenz zu anderen Anforderungen wie z.B. der Wirtschaftlichkeit. In Anbetracht der Dringlichkeit von Klima- und Ressourcenschutz zur Erhaltung unserer Lebensgrundlage, sollte die öffentliche Beschaffung von umweltverträglichen Produkten verpflichtend werden.

- ▶ Dies sollte an zentraler Stelle mit höchster Priorität, beispielsweise in der Bundeshaushaltsordnung und den jeweiligen Länderverordnungen festgeschrieben werden.
- ▶ Zur Vermeidung von softwarebedingter Obsoleszenz sollten von öffentlichen Einrichtungen nur solche Produkte und Produktsysteme beschafft werden dürfen, die Mindestnutzungsdauern (siehe Punkt 3: Gewährleistung einer Mindestnutzungsdauer) gewährleisten können.
- ▶ Die Pflicht zur umweltverträglichen öffentlichen Beschaffung trifft insbesondere auf die Beschaffung von Informationstechnik im Rahmen des „Digitalpakts“ zur Ausstattung von Bildungseinrichtungen mit Computern und Tablets, aber auch für die Digitalisierung von öffentlichen Verwaltungen insgesamt und die dazu verwendeten Haushaltsmittel zu. Die Verwendung dieser Digitalisierungsmittel sollte grundsätzlich an Umwelt- und Langlebigekeitsanforderungen an die Produkte gekoppelt werden.
- ▶ Softwarebetriebene Geräte, für die keine ambitionierte Mindestnutzungsdauer gewährleistet wird, sollten unter den nicht mehr zu beschaffenden Produkten in der Allgemeine Verwaltungsvorschrift der Bundesregierung zur Beschaffung klimafreundlicher Leistungen (AVV Klima) in Anlage 1 gelistet werden.

13. Förderung und Lehre nachhaltiger Softwareentwicklung

- ▶ Der große Vorteil von Software ist, dass sie nicht zwangsläufig an gerätespezifische Beschränkung gebunden ist. Ganz im Gegenteil, anders als fest verdrahtete Logikbausteine, kann sie Maschinen universell und an neue Aufgabenstellungen anpassbar machen („Turingmaschine“). Software kann daher insbesondere auch zur Verlängerung von Produktnutzungsdauern beitragen.
- ▶ Um dies wieder stärker in die Praxis zu bringen, sollte die Entwicklung von nachhaltiger Software und Software, die die Produktlebensdauer verlängert, gefördert werden. Dies kann auf der Ebene von Hochschulen erfolgen (Integration in die Lehre), auf der Ebene von öffentlicher Auftragsvergabe (Entwicklung ressourcenschonender Anwendungen) sowie auf europäischer Ebene bei der Schaffung einer europäischen Cloud-Infrastruktur (GAIA-X).
- ▶ Öffentlich geförderte Software sollte grundsätzlich frei zugänglich und quelloffen sein, damit sie eine möglichst breite Wirkung entfalten kann.

8.4 Steckbriefe

Diesem Dokument sind im Anhang B zusammenfassende Steckbriefe zu den politischen Handlungsempfehlungen beigelegt.

8.5 Zusammenfassung

Die verschiedenen Untersuchungen im Projekt haben deutlich gemacht, dass Software einen wesentlichen Einfluss auf die Nutzungsdauer von vernetzten und mikroelektronischen Produkten hat. Softwarebedingte Obsoleszenz führt zu einem erhöhten Aufkommen an Elektronikschrott, einem erhöhten Ressourcenverbrauch zur Herstellung von Neugeräten und zu wirtschaftlichen Belastungen für die Anwender*innen bzw. Konsument*innen solcher Produkte. Einer softwarebedingten Obsoleszenz sollte deshalb politisch entgegengesteuert werden. Ziel von produktpolitischen Maßnahmen ist es in diesem Fall, eine möglichst lange Nutzungsdauer von Produkten zu ermöglichen. Die Maßnahmen sollten sowohl die Hersteller

und Händler von Produkten adressieren, aber auch die für die Funktionalität der Produkte erforderlichen Dienstleister (z.B. Cloud-Service-Anbieter, Software-Entwickler) und die Verbraucher*innen selbst, die letztlich über die Entsorgung und den Neukauf von Produkten entscheiden.

Die softwarebedingte Einflussnahme auf die Verkürzung der Nutzungsdauer von Produkten wird bisher nur in einigen Facetten und insbesondere von unterschiedlichen regulatorischen Vorgaben adressiert. Die Warenkaufrichtlinie (EU 2019/771) stellt bereits Anforderungen an Informationspflichten und den Erhalt der Vertragsmäßigkeit von Produkten sowie dessen Qualität, was auch die Haltbarkeit und Bereitstellung von Software umfasst. Zudem ist in Deutschland ein Gewährleistungszeitraum von zwei Jahren festgelegt. Die Ökodesign-Richtlinie (RL (EU) 2009/125/EG) legt Mindestanforderungen an den Marktzugang für energieverbrauchsrelevante Produkte fest und enthält für einige Produkte bereits konkrete Anforderungen an die Mindestverfügbarkeit von Soft- und Firmware sowie den Zugang für reparaturrelevante Informationen.

Softwarebedingte Obsoleszenz kann in all ihren Aspekten regulatorisch eingedämmt werden, da die Ursachen meist organisatorischer und wirtschaftlicher Art sind und nicht technisch-physikalischer Art. Die nachfolgenden Empfehlungen richten sich nicht nur an die nationale Gesetzgebung. Da viele Akteure global agieren, sollte mindestens EU-weit ein level-playing field geschaffen werden.

Die zwölf empfohlenen Handlungsempfehlungen sind nach den drei Marktmechanismen Mindestanforderungen für den Marktzugang, Transparenzmaßnahmen zur Förderung des Wettbewerbs und Förderung von Öko-Innovationen gruppiert. Transparenzmaßnahmen sind dabei von besonderer Bedeutung, da sie nicht nur die Beschaffung und Kund*innen befähigen, eine informierte ökologische Entscheidung zu treffen, sondern auch elementarer Baustein sind, auf dem die anderen beiden Maßnahmenpakete aufbauen können. Zudem begeben Transparenzmaßnahmen in der Umsetzung meist weniger Barrieren als Mindestanforderungen.

Die **Mindestanforderungen für den Marktzugang** umfassen:

1. die Möglichkeit zum Betrieb der Geräte ohne externe Abhängigkeiten,
2. die Bereitstellung von sicherheitsrelevanten Software-Updates über einen Mindestzeitraum von 10 Jahren,
3. die Gewährleistung einer Mindestnutzungsdauer von typischerweise 10 Jahren,
4. das Verbot von softwareseitigem Verhindern von Reparatur und Betrieb,
5. Anforderungen an die Kompatibilität und Interoperabilität von Produktsystemen,
6. Anforderungen an externe Dienstleistungen zur Aufrechterhaltung ihrer Dienste für mindestens 10 Jahre sowie
7. Anforderungen an Verkaufs- und Vertriebsplattformen von Software/Apps

Die **Transparenzmaßnahmen zur Förderung des Wettbewerbs** werden gewährleistet durch:

8. verpflichtende Nennung der Abhängigkeiten von softwarebetriebenen Produkten,
9. verpflichtende Angabe des garantierten Supportzeitraums und daraus resultierende Rechtsansprüche,
10. Kennzeichnung von Produkten zu ihrer Reparierbarkeit sowie
11. verpflichtende Nennung der übertragenen Datenmengen und Art der Übertragungsinhalte.

Öko-Innovationen werden gefördert durch:

12. eine verpflichtende umweltverträgliche öffentliche Beschaffung und
13. die Förderung und Lehre nachhaltiger Softwareentwicklung.

Die im Projekt erarbeiteten Handlungsempfehlungen können sowohl in bereits existierende Gesetze integriert werden als auch z.B. im Rahmen des europäischen Circular Economy Action Plans (CEAP) in einer eigenen Verordnung oder Richtlinie münden.

9 Weitere Forschungsbedarfe

Unter Verwendung vielfältiger Methoden hat diese Studie beispielhaft Probleme softwarebedingter Obsoleszenz definiert, erklärt und mögliche Lösungswege aufgezeigt. Dennoch kann sie weder in der Auswahl der Fallbeispiele noch in den Wirkmechanismen und Lösungspfaden einen Anspruch auf Vollständigkeit erheben. Im Folgenden soll daher der weitere Forschungsbedarf skizziert werden.

Grundlagenforschung:

- ▶ Vertiefende Recherchen und Analysen zu kontextverwandten Konzepten und Begriffen wie beispielsweise der „software evolution“ und dem „software aging“ könnten in Zukunft zu einer besseren Operationalisierung und Formalisierung beitragen. In diesem Zusammenhang sollte eine weitere Auswertung von bestehenden Standards und Modellen stattfinden. Insbesondere im industriellen Bereich gibt es bereits eine Vielzahl an Scoring-Modellen und Bewertungssystemen im Softwareentwicklungsbereich, von dem weitere Regulierungsvorhaben profitieren können. Es sollten Vorschläge entwickelt und Bedarfsklärungen mit den (inter-) nationalen Gremien vorgenommen werden.
- ▶ In dieser Studie konnte nicht gezeigt werden, wie softwarebedingte Obsoleszenz mit anderen Spielarten von Obsoleszenz derselben Produktgruppen wechselwirkt und ob gegebenenfalls Synergieeffekte politischer Maßnahmen entstehen. In diesem Kontext sollten insbesondere globale Lieferketten von Chips und Bereitstellung von Treibersoftware untersucht werden.

Folgenabschätzung:

- ▶ Ein nachfolgendes Forschungsvorhaben könnte den Versuch unternehmen, die Umwelteffekte und wirtschaftlichen Folgekosten, sowie Externalisierungs- und Verteilungseffekte softwarebedingter Obsoleszenz zu quantifizieren.
- ▶ Weiterhin sollte der Frage nachgegangen werden, ob Unternehmen ökonomische Vorteile erwachsen können, wenn sie Obsoleszenz vermeiden, oder ob ein grundsätzlicher Zielkonflikt zwischen Ökonomie und Ökologie im Design lebenslang haltbarer Produkte besteht. Darauf aufbauend stellt sich die Frage, wie Gesellschaft und Regierungen angesichts multipler ökologischer Katastrophen mit diesem Zielkonflikt umgehen.

Politisch und rechtlicher Rahmen:

- ▶ Im Rahmen der vorliegenden Studie war es nicht möglich, eine genaue rechtliche Bewertung und Sammlung von Fällen von Täuschung der Kund*innen und Wettbewerbsverzerrung im Kontext softwarebedingter Obsoleszenz vorzunehmen.
- ▶ Es konnte nicht abschließend geklärt werden, welche Pflichten aus der Warenverkaufslinie (EU 2019/771) erwachsen. Dafür wäre eine Definition und Auslegung von Verkehrssitte und von vernünftigerweise erwartbaren Qualitätsanforderungen nötig.
- ▶ Eine spannende Frage bleibt, inwieweit die politischen Maßnahmen einzeln und in welcher Konstellation des zeitlichen Inkrafttretens welche Wirksamkeit entfalten.

Des Weiteren soll diese Studie einen Beitrag leisten, Entscheidungsträger*innen und Öffentlichkeit für die ökologischen Folgen der Digitalisierung zu sensibilisieren. Auch hierbei besteht noch Forschungs- und Handlungsbedarf. Offene Fragen sind u.a., wie die bestehende Transformationsforschung im Bereich der Digitalisierung auch für das Thema

Softwareobsoleszenz sensibilisiert und wie die Thematik in anderen Politikfelder verankert werden kann. Außerdem sollten anschlussfähige Narrative für nicht software-affine Zielgruppen entwickelt werden.

10 Quellenverzeichnis

- Abdullah, R.; Abdullah, S.; Din, J.; Tee, M.; others (2015): A Systematic Literature Review of Green Software Development in Collaborative Knowledge Management Environment. In: International Journal of Advanced Computer Technology (IJACT) 9.
- acatech (2011): Cyber-Physical Systems. Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion. München: acatech – Deutsche Akademie der Technikwissenschaften, 2011.
- AGCM (Autorità Garante Della Concorrenza e del Mercato) (2018): Apple and Samsung fined for software updates that have caused serious troubles and/or have reduced functionality of some mobile phones. (Pressemitteilung vom 24.10.2018), <https://en.agcm.it/en/media/press-releases/2018/10/PS11009-PS11039> (25.10.2018).
- Albers, E. (2015): Software im Kontext der Nachhaltigkeit, In: Gröger, J.; Köhn, M. (2015): Nachhaltige Software, Dokumentation des Fachgesprächs "Nachhaltige Software" am 28.11.2014 (Dokumentationen 07/2015), Dessau-Roßlau: Umweltbundesamt, S. 29-37.
- Apple (2020): About battery health management in Mac notebooks. Online Article at April 16, 2020, URL: <https://support.apple.com/en-us/HT211094> (29.04.2020).
- Apple (2021a): Apple Developer Program. <https://developer.apple.com/programs/how-it-works/> (28.07.2021).
- Apple (2021b): Corona-Warn-App im App Store. <https://apps.apple.com/de/app/corona-warn-app/id1512595757> (21.09.2021).
- Apple (2022): iPhone Battery and Performance. Understand iPhone performance and its relation to your battery. <https://support.apple.com/en-us/HT208387> (10.02.2022).
- Apple; Google (2020): Exposure Notification. Cryptography Specification. <https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ExposureNotification-CryptographySpecificationv1.2.pdf>.
- Baker, T. (2015): Industrie-Report „Smart Home“. https://www.bakertilly.de/uploads/media/Studie_Smart-Home_online_01.pdf (08.12.2020).
- Balta-Ozkan, N.; Boteler, B.; Amrighi, O. (2014): European smart home market development: Public views on technical and economic aspects across the United Kingdom, Germany and Italy. Energy Research & Social Science (2014) 65-77.
- Balzert, H. (2009): Anforderungen und Anforderungsarten. In: Lehrbuch der Softwaretechnik: Basiskonzepte und Requirements Engineering. Spektrum Akademischer Verlag, Heidelberg, S.455-474.
- Bartels, B.; Ermel, U.; Sandborn, P.; Pecht, M. (2012): Strategies to the Prediction, Mitigation, and Management of Product Obsolescence (Reihe: Wiley Series in Systems Engineering and Management 87), Hoboken/New Jersey: Wiley.
- Bartels, B.; Poppe, E. (2019). Obsoleszenz als Managementthema. In: Erik Poppe/Jörg Longmuß (Eds.), Geplante Obsoleszenz (123-142). Bielefeld: transcript Verlag. <https://doi.org/10.14361/9783839450048-007>.
- Belkin (2014): NETCAM HD+. User Manual. http://cache-www.belkin.com/support/dl/F7D7602V2_8820ed01325uk_RevC00_NetCam.pdf (07.12.2021).
- Belkin (2019): Cant create Belkin Netcam account. <https://community.belkin.com/s/question/0D52T00004uYejdSAC/cant-create-belkin-netcam-account> (30.09.2020).
- Belkin (2020): Wemo NetCam Ende der Lebensdauer Mitteilung. <https://www.belkin.com/de/support-article?articleNum=316642> (12.05.2020).
- Bisset, J. (2018): Apple fined \$6.6M in Australia after Error 53 controversy, Online Article at cnet.com from 2018-06-18, URL: <https://www.cnet.com/news/apple-bricked-our-phones-with-error-53-now-it-owes-6-8-million-in-australia/> (28.04.2020).

Bitdefender (2016): Security Awareness in the Age of Internet of Things.
<https://download.bitdefender.com/resources/files/News/CaseStudies/study/139/Bitdefender-Whitepaper-IoTSecurity-A4-en-final.pdf> (29.10.2021).

Blauer Engel (2020): Ressourcen- und energieeffiziente Softwareprodukte, RAL gGmbH: Bonn, DE-UZ 215 Ausgabe Januar 2020.

BMU (2020): Umweltpolitische Digitalagenda, Berlin: Bundesministerium für Umwelt, Naturschutz und nukleare Sicherheit.
https://www.bmu.de/fileadmin/Daten_BMU/Pool/Broschueren/broschuere_digitalagenda_bf.pdf (17.12.2021).

BMWi (2016): Digitale Strategie 2025. Berlin, Bundesministerium für Wirtschaft und Energie (BMWi).

BMWi (2017): Weißbuch Digitale Plattformen, Digitale Ordnungspolitik für Wachstum, Innovation, Wettbewerb und Teilhabe. Berlin, Bundesministerium für Wirtschaft und Energie (BMWi).

Bohlen, J.M.; George, M. (1957): The diffusion process. Special Report 24, Iowa State College.
https://lib.dr.iastate.edu/specialreports/24/?utm_source=lib.dr.iastate.edu%2Fspecialreports%2F24&utm_medium=PDF&utm_campaign=PDFCoverPages (07.12.2021).

Bowlds, T.F.; Fossaceca, J.M.; Iammartino, R. (2018): Software obsolescence risk assessment approach using multicriteria decision-making. *Systems Engineering*. 2018; 21: 455– 465. <https://doi.org/10.1002/sys.21446>.

Breitkopf, A. (2019): Verteilung der Smart-Home-Geräte in Deutschland 2019. In: Statista: URL:
<https://de.statista.com/statistik/daten/studie/1067828/umfrage/verteilung-der-smart-home-geraete-in-deutschland/> (17.01.2019).

BSI (2020a): Herstelleranforderungen zur Sicherheit von Smartphones. Bundesamt für Sicherheit in der Informationstechnik (BSI): Bonn.

BSI (2020b): BSI entwickelt Diskussionsgrundlage zu Sicherheitsanforderungen für Smartphones. Pressemitteilung des Bundesamt für Sicherheit in der Informationstechnik (BSI): Bonn (25.02.2020).
https://www.bsi.bund.de/DE/Presse/Pressemitteilungen/Presse2020/Sicherheitsanforderungen-Smartphones_250220.html (22.04.2020).

Bundesgesetzblatt (Hg.) (2017): Gesetz über die Bereitstellung von Funkanlagen auf dem Markt (Funkanlagengesetz - FuAG). Online verfügbar unter
https://www.bgbl.de/xaver/bgbl/start.xav?start=%2F%2F%5B%40attr_id%3D%27bgbl117s1947.pdf%27%5D#__bgbl_%2F%2F%5B%40attr_id%3D%27bgbl117s1947.pdf%27%5D__1641822142528, zuletzt aktualisiert am 10.01.2022 (10.01.2022).

Bundesministerium für Wirtschaft und Energie (2016): SmartHome2Market Marktperspektiven für die intelligente Heimvernetzung – 2016. https://www.digitale-technologien.de/DT/Redaktion/DE/Downloads/Publikation/smarthome-broschuere.pdf?__blob=publicationFile&v=9 (29.10.2021).

Burrus, P. F.; Jones, R.; Lew, C. (2012): A commercial approach to military system sustainment. 2012 IEEE AUTOTESTCON Proceedings, Anaheim, CA, 2012, pp. 166-168.

Chiramal, M. (2012): Application of commercial-grade digital equipment in nuclear power plant safety systems. Proceedings 20th IEEE Symposium on Reliable Distributed Systems, New Orleans, LA, USA, 2001, pp. 176-178.

Cotroneo, D.; Natella, R.; Pietrantuono, R.; Russo, S. (2014). A Survey of Software Aging and Rejuvenation Studies. *ACM Journal on Emerging Technologies in Computing Systems*. 10. 10.1145/2539117.

CWA Open Source Projekt (2021): Sicherheitsupdate für Corona-Warn-App-Server.
<https://www.coronawarn.app/de/blog/2020-11-19-security-update/> (04.08.2021).

Dannoritzer, C. (Produzentin und Regisseurin) (2010): The Light Bulb Conspiracy [Film]. Frankreich: o.O.

Dano, E. B. (2019): Importance of Reuse and Modularity in System Architecture, 2019 International Symposium on Systems Engineering (ISSE), Edinburgh, United Kingdom, 2019, pp. 1-8.

- David (2018): Philips Hue: Unterschiede der 1. bis 6. Generation. In: *Smartlights.de - Dein Blog über Philips Hue & smarte Beleuchtung*, 22.05.2018. Online verfügbar unter <https://www.smartlights.de/licht/philips-hue/unterschiede-und-vergleich-der-philips-hue-generationen/> (10.01.2022).
- Deloitte (2018): Smart Home Consumer Survey 2018 – Ausgewählte Ergebnisse für den deutschen Markt. https://www2.deloitte.com/content/dam/Deloitte/de/Documents/technology-media-telecommunications/Deloitte_TMT_Smart_Home_Studie_18.pdf (22.01.2020).
- Destatis (2020): Entwicklung der Privathaushalte bis 2040 - Ergebnisse der Haushaltsvorausberechnung 2020. <https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Bevoelkerung/Haushalte-Familien/Publikationen/Downloads-Haushalte/entwicklung-privathaushalte-5124001209004.html> (29.10.2021).
- DG CCRF (2020, February 2020): Ralentissement du fonctionnement de certains iPhone : une enquête de la DGCCRF conduit au paiement d'une amende transactionnelle de 25 M€ par le groupe Apple. Pressemitteilung der Direction générale de la concurrence, de la consommation et de la répression des frauds (DG CCRF): Paris. <https://www.economie.gouv.fr/dgccrf/ralentissement-du-fonctionnement-de-certains-iphone-une-enquete-de-la-dgccrf-conduit-au> (22.04.2020).
- Dziuk, M. A. (2014): Using FPGA as synthetic instrumentation in automated test sets: Benefits and examples. 2014 IEEE AUTOTEST, St. Louis, MO, 2014, pp. 136-141.
- EC (European Commission) (2020a): Clarifications on the initiative of upload of software into radio equipment. 2020/04/2020. <https://ec.europa.eu/docsroom/documents/40822> (20.10.2020).
- EC (European Commission) (2020b): Circular Economy Action Plan. For a cleaner and more competitive Europe. https://ec.europa.eu/environment/circular-economy/pdf/new_circular_economy_action_plan.pdf (28.08.2020).
- Epson (2020): Error message "Parts inside the printer have reached the end of their service life". <https://www.epson.de/en/viewcon/corporatesite/kb/index/1582> (28.04.2020).
- Evehome (2021): HomeKit über Thread | evehome.com. <https://www.evehome.com/de/thread> (10.08.2021).
- Fairphone (2017, July 20): Why we had to stop supporting the Fairphone 1. <https://www.fairphone.com/de/2017/07/20/why-we-had-to-stop-supporting-the-fairphone-1/> (29.04.2020).
- Gardner, D. A. F. (2016): Legacy ATE upgrade: Lessons learned, 2016 IEEE AUTOTESTCON, Anaheim, CA, 2016, pp. 1-5.
- Geisberger, E.; Broy, M. (Hrsg.) (2012): agendaCPS. Integrierte Forschungsagenda Cyber-Physical Systems. München: acatech STUDIE 03/2012. https://www.acatech.de/wp-content/uploads/2018/03/acatech_STUDIE_agendaCPS_Web_20120312_superfinal.pdf (04.05.2020).
- Gerstl, S. (2019): Software-Wartbarkeit trotz Erosion. In: *embedded-software engineer*. <https://www.embedded-software-engineering.de/software-wartbarkeit-trotz-erosion-a-824308/> (26.07.2019).
- Giga (2017): iOS-App programmieren - Voraussetzungen und Tipps. In: *Giga*, 29.05.2017. <https://www.giga.de/downloads/ios-11/tipps/ios-app-programmieren-voraussetzungen-und-tipps/> (28.07.2021).
- GitHub (2021): GitHub - corona-warn-app/cwa-documentation: Project overview, general documentation, and white papers. <https://github.com/corona-warn-app/cwa-documentation> (29.07.2021).
- Google (2021): Corona-Warn-App – Apps bei Google Play. <https://play.google.com/store/apps/details?id=de.rki.coronawarnapp> (21.09.2021).
- Google (2022): Softwareupdates auf Google Pixel. <https://support.google.com/nexus/answer/4457705?hl=de#zippy=%2Cpixel-und-pixel-pro%2Cpixel-a-a-xl-xl-a-a-g-und-a-g%2Cpixel-und-%C3%A4ltere-modelle>

Graser, F. (2015). Software-Erosion lässt sich nun schneller aufspüren. In: Elektronik Praxis. <https://www.elektronikpraxis.vogel.de/software-erosion-laesst-sich-nun-schneller-aufspueren-a-497392/> (26.07.2019).

Gröger, J. et al. (2018): Entwicklung und Anwendung von Bewertungsgrundlage für ressourceneffiziente Software unter Berücksichtigung bestehender Methodik (Texte 105/2018), Dessau-Roßlau: Umweltbundesamt. https://www.umweltbundesamt.de/sites/default/files/medien/1410/publikationen/2018-12-12_texte_105-2018_ressourceneffiziente-software_0.pdf (24.07.2019).

Gröger, J.; Köhn, M. (2015): Nachhaltige Software, Dokumentation des Fachgesprächs "Nachhaltige Software" am 28.11.2014 (Dokumentationen 07/2015), Dessau-Roßlau: Umweltbundesamt.

Grottke, M.; Matias Jr. R.; Trivedi, K. (2008): The fundamentals of software aging. Software Reliability Engineering Workshops, 2008. ISSRE Wksp 2008. IEEE International Conference on. 1 - 6. 10.1109/ISSREW.2008.5355512.

GSMA (2012): Vision of Smart Home The Role of Mobile in the Home of the Future. <https://www.gsma.com/iot/wp-content/uploads/2012/03/vision20of20smart20home20report.pdf> (29.10.2021).

Hahn, F. (2020): Softwarebedingte Obsoleszenz. Eine Untersuchung der Ursache-Wirkungszusammenhänge auf Basis von Fallstudien (Masterarbeit). Technische Universität Berlin, pp.51.

Hardenburg, G.L. (1999): Successful migration of a legacy system to an open architecture. 1999 IEEE AUTOTESTCON Proceedings (Cat. No.99CH36323), San Antonio, TX, USA, 1999, pp. 405-410.

Harnack, J. (2012): Life cycle planning from product development to long term sustainment. 2012 IEEE AUTOTESTCON Proceedings, Anaheim, CA, 2012, pp. 29-33.

Heagney, C. P.; Walker, L.J. (2018): Virtual Applications Reduce Cyber Attack Surface for Test Program Sets and Station Software. 2018 IEEE AUTOTESTCON, National Harbor, MD, 2018, pp. 1-8.

Heinz, R.; Meyer, K. (2020): Der französische Reparaturindex. Ein Modell für Deutschland und die EU? RTR Factsheet, Runder Tisch Reparatur: Berlin. <https://runder-tisch-reparatur.de/wp-content/uploads/2020/04/RTR-Factsheet-Reparaturindex-April-2020.pdf> (22.04.2020).

Heise Online (2008, January 01): Der Schrottzähler - Wie Epson seine Drucker einfach abschaltet. <https://www.heise.de/multimediateil/der-schrottzähler-wie-epson-seine-drucker-einfach-abschaltet-1507074.html> (28.04.2020).

Herbig, D. (2020a) (2020, March 03): Smarte Leuchten: Osram schaltet die Lightify-Cloud ab. Heise Online. <https://www.heise.de/newsticker/meldung/Smarte-Leuchten-Osram-schaltet-die-Lightify-Cloud-ab-4679538.html> (09.12.2021).

Herbig, D. (2020b) (2020, March 06): Sonos stampft umstrittenen Recycling-Modus ein. Heise Online. <https://www.heise.de/newsticker/meldung/Sonos-stampft-umstrittenen-Recycling-Modus-ein-4677266.html> (29.04.2020).

Hess, Christian (2018): Geplante Obsoleszenz: Rechtliche Zulässigkeit in der Lebensdauerplanung von technischen Gebrauchsgütern. Baden-Baden: Nomos Verlagsgesellschaft mbH & Co. KG.

Hilty, L.M.; Naumann, S.; Kern, E.; Filler, A.; Guldner, A.; Gröger, J. (2017): Kriterienkatalog für nachhaltige Software, URL: https://www.umwelt-campus.de/fileadmin/Umwelt-Campus/Greensoft/Kriterienkatalog_nachhaltige_Software_v01_2017-05-31.pdf (24.07.2019)

Hilty, Lorenz; Lohmann, Wolfgang; Behrendt, Siegfried; Evers-Wölk, Michaela; Fichter, Klaus; Hintemann, Ralph (2015): Grüne Software. Schlussbericht zum Vorhaben: Ermittlung und Erschließung von Umweltschutzpotenzialen der Informations- und Kommunikationstechnik (Green IT). Studie im Auftrag des Umweltbundesamtes, Berlin, Förderkennzeichen 3710 95 302/3.

IEEE (1998): IEEE Standard for a Software Quality Metrics Methodology. In: IEEE Std 1061-1998, vol., no., pp.i-, 31 Dec. 1998.

ISO/IEC 25000:2014 (2014): Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE.

iSpy (2020): Belkin IP camera. <https://www.ispyconnect.com/man.aspx?n=Belkin> (30.09.2020).

Jaeger-Erben, M.; Hipp, T.; Nachwuchsgruppe Obsoleszenz (Hrsg., 2017): Letzter Schrei oder langer Atem? – Erwartungen und Erfahrungen im Kontext von Langlebigkeit bei Elektronikgeräten. Deskriptive Auswertung einer repräsentativen Online-Befragung in Deutschland. OHA-Texte 1/2017.

<https://challengeobsolescence.info/aktuelles/letzter-schrei-oder-langer-atem> (Abruf am 28.08.2020).

Jaeger-Erben, Melanie (2019): Eine Frage der Kultur? Gesellschaftliche Treiber von Obsoleszenz. In: Erik Poppe/Jörg Longmuß (Eds.), Geplante Obsoleszenz (171-190). Bielefeld: transcript Verlag.

<https://doi.org/10.14361/9783839450048-010>.

Jang, E.; Johnson, M.; Burnell, E.; Heimerl, K. (2017). Unplanned Obsolescence: Hardware and Software After Collapse. 93-101. 10.1145/3080556.3080566.

Kantar Emnid; vzbv (2017). Haltbarkeit und Reparierbarkeit von Produkten. Berichtsgrafiken Mai 2017,

https://www.vzbv.de/sites/default/files/downloads/2017/06/01/umfrage_-

[_haltbarkeit_und_reparierbarkeit_von_produkten_o_gewaehrleistung.pdf](https://www.vzbv.de/sites/default/files/downloads/2017/06/01/umfrage_-haltbarkeit_und_reparierbarkeit_von_produkten_o_gewaehrleistung.pdf) (03.07.2019).

Keimeyer, F.; Geilhofer, P.; Gsell, M.; Graulich, K.; Prakash, S.; Scherf, C-S; Brönneke, T.; Gildeggen, R.; Schmitt, R. (2020): Weiterentwicklung von Strategien gegen Obsoleszenz einschließlich rechtlicher Instrumente (=Texte 115/2020), Dessau-Roßlau: Umweltbundesamt.

[https://www.umweltbundesamt.de/sites/default/files/medien/479/publikationen/texte_115-](https://www.umweltbundesamt.de/sites/default/files/medien/479/publikationen/texte_115-2020_weiterentwicklung_von_strategien_gegen_obsoleszenz_einschliesslich_rechtlicher_instrumente.pdf)

[2020_weiterentwicklung_von_strategien_gegen_obsoleszenz_einschliesslich_rechtlicher_instrumente.pdf](https://www.umweltbundesamt.de/sites/default/files/medien/479/publikationen/texte_115-2020_weiterentwicklung_von_strategien_gegen_obsoleszenz_einschliesslich_rechtlicher_instrumente.pdf) (28.08.2020).

Kenney, M; Pon, B. (2011): Structuring the Smartphone Industry: Is the Mobile Internet OS Platform the Key? In: Journal of Industry, Competition and Trade. 11. 10.2139/ssrn.1851686.

Klaus, J. (2020): Positiv-Ergebnis kommt nicht in der App an. In: Digitales, 19.10.2020.

<https://www.zdf.de/nachrichten/digitales/coronavirus-app-test-ergebnisse-100.html> (22.07.2021).

Kohlick, R. (2020, March 11): Sonos-Lautsprecher: Audio-Hersteller sieht seinen Fehler endlich ein. Giga Online Magazin. <https://www.giga.de/news/sonos-lautsprecher-update-macht-speaker-kaputt-mit-voller-absicht-11-03-2020/> (29.04.2020).

Költzsch, Tobias (2021): Smart Home: Zigbee Alliance mit neuem Namen und neuem Standard - Golem.de.

In: Golem.de, 12.05.2021. <https://www.golem.de/news/smart-home-zigbee-alliance-mit-neuem-namen-und-neuem-standard-2105-156461.html> (05.08.2021).

Kumar, N.; Koley, I.; Krishnamurthy, P. R.; Rao, S. N. (2019): Regulatory review of computer based systems: Indian perspectives. 2010 2nd International Conference on Reliability, Safety and Hazard - Risk-Based Technologies and Physics-of-Failure Methods (ICRESH), Mumbai, 2010, pp. 475-478.

Kunst, Alexander (2019): Umfrage in Deutschland zum Besitz von Smart-Home-Geräten 2019. In: Statista.

<https://de.statista.com/prognosen/999788/umfrage-in-deutschland-zum-besitz-von-smart-home-geraeten> (Abruf am 28.08.2020).

Lambert, F. (2019, October 04): Tesla's software update that resulted in battery capacity loss is under probe by NHTSA. Electrek Online Magazine., <https://electrek.co/2019/10/04/tesla-software-updat-battery-capacity-loss-fire-investigation-nhtsa/> (29.04.2020).

Lambert, F. (2020, March 03): Tesla releases software update boosting displayed range of Model S/X vehicles. Electrek Online Magazine. <https://electrek.co/2020/03/03/tesla-software-update-boosting-range-model-s-x-vehicles/> (29.04.2020).

Lehman M.M. (1996): Laws of software evolution revisited. In: Montangero C. (eds.) Software Process Technology. EWSPT 1996. Lecture Notes in Computer Science, vol 1149. Springer, Berlin, Heidelberg.

Lohmann, W. (2015): Einfluss von Software auf Energie- und Ressourceneffizienz. In: Gröger, J.; Köhn, M. (Hrsg.): Nachhaltige Software, Dokumentation des Fachgesprächs "Nachhaltige Software" am 28.11.2014 (Dokumentationen 07/2015), Dessau-Roßlau: Umweltbundesamt, S. 23-28.

- Lowenstein, D.; LaGrotta, J. (2015): Strategies for extending the life of ATE systems for another 10 or 20 years. 2015 IEEE AUTOTESTCON, National Harbor, MD, 2015, pp. 65-70.
- Martin, C; Garcia-Swartz, D. (2015): From Mainframes to Smartphones: A History of the International Computer Industry [eBook]. Cambridge, Massachusetts; London, England: Harvard University.
- MDR Aktuell (2019, December 20): Regierung einigt sich auf Gesetz gegen Vernichtung von Neuwaren. Mitteldeutscher Rundfunk (MDR) Aktuell. <https://www.mdr.de/nachrichten/politik/inland/vernichtung-neuwaren-retouren-gesetz-100.html> (29.04.2020).
- Moorhead, G. (1993): New computers extend the life of deployed testers. AUTOTESTCON 93, San Antonio, TX, USA, 1993, pp. 667-670.
- Morris, A. T. (2000): COTS Score: an acceptance methodology for COTS software. 19th DASC. 19th Digital Avionics Systems Conference. Proceedings (Cat. No.00CH37126), Philadelphia, PA, USA, 2000, pp. 4B2/1-4B2/8 vol.1.
- Neal, T.M. (2004): Obsolescence replacement-applied technology. Proceedings AUTOTESTCON 2004., San Antonio, TX, USA, 2004, pp. 58-61.
- O'Donnell, S. J.; Krayewsky, M.F. (2006): Hardware Agnostic Approach to TPS Life Cycle Sustainment. 2006 IEEE Autotestcon, Anaheim, CA, 2006, pp. 371-375.
- Oehme, I.; Jacob, A.; Cerny, L.; Fabian, M.; Golde, M.; Krause, S.; Löwe, C.; Unnerstall, H. (2017): Strategien gegen Obsoleszenz. Sicherung einer Produktmindestlebensdauer sowie Verbesserung der Produktnutzungsdauer und der Verbraucherinformation, Dessau-Roßlau: Umweltbundesamt.
- Openhab (2021): tado° - Bindings. <https://www.openhab.org/addons/bindings/tado/> (20.09.2021).
- Parnas, D. L. (1994): Software aging. In: Proceedings, 16th International Conference on Software Engineering. May 16-21, 1994, Sorrento, Italy. 16th International Conference on Software Engineering. IEEE Computer Society. Los Alamitos, CA: IEEE Computer Society Press, S. 279–287.
- Parnas, D. L. (1996): Why software jewels are rare. In: Computer 29 (2), S. 57–60. DOI: 10.1109/2.485847.
- Philips Hue (2021a): Get Started - Philips Hue Developer Program. <https://developers.meethue.com/develop/get-started-2/> (13.08.2021).
- Philips Hue (2021b): End-of-Support-Richtlinie | Philips Hue. <https://www.philips-hue.com/de-at/support/legal/end-of-support-policy> (06.08.2021).
- Philips Hue (2022): Philips Hue White Filament Lampe Einzelpack E27 Globe | Philips Hue. Online verfügbar unter <https://www.philips-hue.com/de-de/p/philips-hue-white-filament-lampe-einzelpack-e27-globe/8718699688882#specifications> (10.01.2022).
- Poppe, E.; Longmuß, J. (2019): Zu Begriff und Theorie der geplanten Obsoleszenz. In: Erik Poppe/Jörg Longmuß (Eds.), Geplante Obsoleszenz (17-38). Bielefeld: transcript Verlag. <https://doi.org/10.14361/9783839450048-003>.
- Prakash, S.; Dehoust, G.; Gsell, M.; Schleicher, T.; Stamminger, R. (2016): Einfluss der Nutzungsdauer von Produkten auf ihre Umweltwirkung: Schaffung einer Informationsgrundlage und Entwicklung von Strategien gegen »Obsoleszenz« (=Texte 11/2016), Dessau-Roßlau: Umweltbundesamt. www.umweltbundesamt.de/sites/default/files/medien/378/publikationen/texte_11_2016_einfluss_der_nutzungsdauer_von_produkten_obsoleszenz.pdf (05.03.2019), S. 315–318.
- Radclyffe, C. (2020, April 29): Belkin May Never Be Trusted Again After This Story. <https://www.forbes.com/sites/charlesradclyffe/2020/04/29/belkin-may-never-be-trusted-again-after-this-story/?sh=626995275795> (09.12.2021).
- Reuters (2020): Cobol Blues (Grafische Übersicht zur Verwendung von Cobol). <http://fingfx.thomsonreuters.com/gfx/rngs/USA-BANKS-COBOL/010040KH18J/index.html> (25.04.2020).

- Reuters (2021, Juli 30): Tesla agrees to pay \$1.5 mln to settle claims over battery voltage reduction. <https://www.reuters.com/business/autos-transportation/tesla-agrees-pay-15-mln-settle-claims-over-temporary-battery-voltage-reduction-2021-07-29/> (10.02.2022).
- Robert Koch-Institut (2020): Corona Warn-App. Bericht zur Datenschutz-Folgenabschätzung für die Corona-Warn-App der Bundesrepublik Deutschland. Version 1.0.1 (18.06.2020).
- Robert Koch-Institut (2021): Kennzahlen zur CWA. https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/WarnApp/Archiv_Kennzahlen/Kennzahlen_19082021.pdf?__blob=publicationFile (07.12.2021).
- Robinson, B.; Hughes, B.; Bauer, R.; Harnack, J. (2015): Proactively managing obsolescence with test system architecture. 2015 IEEE AUTOTESTCON, National Harbor, MD, 2015, pp. 87-92.
- Rojo, FJR; Shehap, E.; Cheruvu, E.; Blackman, I.; Rumney, GA (2010): Key Challenges in Managing Software Obsolescence for Industrial Product-Service Systems (IPS2). In: Proceedings of 2nd CIRP international conference on industrial product-service systems, Linköping, 14-15 Apr 2010, pg. 393-398.
- Rother, B. (2012, May 18): Runcore Invincible: Selbstzerstörende SSD. PC-Magazin. <https://www.pc-magazin.de/news/runcore-invincible-selbstzerstoerende-ssd-1281882.html> (29.04.2020).
- Rowe, Martin (2017, August 10): IoT: The Inference of Things. edn.com. <https://www.edn.com/iot-the-interference-of-things/> (20.10.2020).
- Samsung (2020): Mein Smartphone/Tablet ist langsam oder hängt, was kann ich prüfen? <https://www.samsung.com/de/support/mobile-devices/mein-smartphone-ist-langsam-oder-hangt-was-kann-ich-pruefen/> (28.04.2020).
- Sandborn, P. (2007): Editorial Software Obsolescence—Complicating the Part and Technology Obsolescence Management Problem. Components and Packaging Technologies, IEEE Transactions on. 30. 886-888. 10.1109/TCAPT.2007.910918.
- Sandborn, P. A.; Prabhakar, V.J. (2015): The Forecasting and Impact of the Loss of Critical Human Skills Necessary for Supporting Legacy Systems. In IEEE Transactions on Engineering Management, vol. 62, no. 3, pp. 361-371, Aug. 2015.
- Santarius, T. (2015): Der Rebound-Effekt. Ökonomische, psychische und soziale Herausforderungen der Entkopplung von Energieverbrauch und Wirtschaftswachstum. Wirtschaftswissenschaftliche Nachhaltigkeitsforschung Band 18. Marburg: Metropolis, 2015.
- Schäfer, A. (2016, June 23): iPhone Fehler 53: US-Gericht weist Klage ab. <https://www.connect.de/news/iphone-6-6s-fehler-53-sammelklage-usa-gescheitert-3196030.html> (28.04.2020).
- Schlacke, S; Tonner, K.; Gawel, E.; Alt, M.; Bretschneider, W. (2015): Stärkung eines nachhaltigen Konsums im Bereich Produktnutzung durch Anpassungen im Zivil- und öffentlichen Recht (=Texte 72/2015), Dessau-Roßlau: Umweltbundesamt. https://www.umweltbundesamt.de/sites/default/files/medien/378/publikationen/texte_72_2015_staerkung_eines_nachhaltigen_konsums_im_bereich_produktnutzung_0.pdf (22.04.2020).
- Schridde, S.; Kreiß, C. (2013): Geplante Obsoleszenz: Entstehungsursachen, Konkrete Beispiele, Schadensfolgen, Handlungsprogramm. Gutachten im Auftrag der Bundestagsfraktion Bündnis 90 / Die Grünen, Berlin: ARGE REGIO Stadt- und Regionalentwicklung GmbH.
- SecurityLab (2018): Global security update availability for Smartphones (Januar/Februar 2018 Report) [Abbildung], <https://twitter.com/SecX13/status/968225118517452800/photo/1> (29.04.2020).
- Sparr, C. J.; Fox, R. A.; Song, Y. B. (2018): Optimizing Regression Testing of Software for the Consolidated Automated Support System. 2018 IEEE AUTOTESTCON, National Harbor, MD, 2018, pp. 1-5.
- Statista (2019): Absatz von Smartphones in Deutschland in den Jahren 2009 bis 2019. <https://de.statista.com/statistik/daten/studie/77637/umfrage/absatzmenge-fuer-smartphones-in-deutschland-seit-2008/> (08.12.2020)

- Statista (2019): Deutschlands Haushalte werden immer smarter. <https://www.energie-klimaschutz.de/smarte-zuhause-sicht-des-verbraucherschutzes/> (29.10.2021)
- Statista (2019): Smart Home – Deutschland. URL: <https://de.statista.com/outlook/279/137/smart-home/deutschland#market-users> (17.01.2019).
- Statista (2020): Deutschlands Haushalte werden immer smarter. <https://de.statista.com/infografik/3105/anzahl-der-smart-home-haushalte-in-deutschland/> (08.12.2020)
- Statista (2020): Smartphone market share by vendor in Europe. <https://www.statista.com/statistics/632599/smartphone-market-share-by-vendor-in-europe/> (08.12.2020)
- Statista (2021): Smartphone market in Europe - Statistics & Facts. <https://www.statista.com/topics/3341/smartphone-market-in-europe/#dossierKeyfigures> (29.10.2021)
- Statistisches Bundesamt (2019): 9% Der Nutzer*innen und-nutzer verwenden Smart-Home-Technologien. https://www.destatis.de/DE/Presse/Pressemitteilungen/2019/09/PD19_335_639.html (22.01.2020).
- Steinberg, J. (2020, March 03): Apple’s \$500 Million iPhone Deal Warrants Approval, Plaintiffs Say. Bloomberg Law. <https://news.bloomberglaw.com/product-liability-and-toxics-law/apples-500-million-iphone-deal-warrants-approval-plaintiffs-say> (22.04.2020).
- Stoll, K. (2020a): Senior Programmer händeringend gesucht – für 60 Jahre alte Programmiersprache. In: t3n Online, <https://t3n.de/news/senior-programmer-haenderingend-1271123/?fbclid=IwAR27Qxn9VT8mVxEmWu2M7puRn3V9AAvGmlaknjcu1VEMiY57lkqGt2NFTQ> (25.04.2020).
- Stoll, K. (2020b): Eine kleine Library stürzt das JavaScript-Ökosystem ins Chaos. Online Artikel vom 01.05.2020, In: t3n Online. <https://t3n.de/news/kleine-library-stuerzt-ins-chaos-1274543/> (01.05.2020).
- Tacha, N.; McCarthy, A.; Powell, B.; Veeramani, A. (2010): How to mitigate hardware obsolescence in NextGen test systems. In IEEE Aerospace and Electronic Systems Magazine, vol. 25, no. 7, pp. 30-36, July 2010.
- Taentzer, G., Goedicke, M., Paech, B., Schneider, K., Schürr, A., & Vogel-Heuser, B. (2019). The Nature of Software Evolution. Managed Software Evolution, 9–20. doi:10.1007/978-3-030-13499-0_2.
- The Recycler (2019, October 22): Epson in class action lawsuit for firmware updates. <https://www.therecycler.com/posts/epson-in-class-action-lawsuit-for-firmware-updates/> (28.04.2020).
- Toku, H. A. (2013): Developing new Automatic Test Equipments (ATE) using systematic design approaches. 2013 IEEE AUTOTESTCON, Schaumburg, IL, 2013, pp. 1-7.
- Torres, R.; Bencomo, N.; Astudillo, H. (2013): Addressing the QoS drift in specification models of self-adaptive service-based systems, 2013 2nd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE), San Francisco, CA, 2013, pp. 28-34.
- Trimmer, K. J.; Blanton, J. E.; Schambach, T. (1998): An evaluation of factors affecting professional obsolescence of information technology professionals. Proceedings of the Thirty-First Hawaii International Conference on System Sciences, Kohala Coast, HI, USA, 1998, pp. 406-414 vol.6.
- Trust, A. (2021, May 12): Matter: Neuer Smart-Home-Standard von Apple, Amazon, Google und Co. <https://macnotes.de/2021/05/12/matter-neuer-smart-home-standard-von-apple-amazon-google-und-co/> (05.08.2021).
- Tsikalas, G. (1998): Supplier's perspective on commercial air transport avionics business practices, 17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference. Proceedings (Cat. No.98CH36267), Bellevue, WA, USA, 1998, pp. A11-1.
- Tukey, John W. (1958): The Teaching of Concrete Mathematics. In: The American Mathematical Monthly. Vol. 65, no. 1, Jan. 1958, S. 2.
- Voas, J.; Agrest, W.W. (2004): Software quality from a behavioral perspective. IEEE Journal of IT Professional.

VZBV (2019, May 21): Verbraucher wünschen sich fünf Jahre lang Smartphone-Updates. Verbraucherzentrale Bundesverband (VZBV), Pressemitteilung. <https://www.vzbv.de/pressemitteilung/verbraucher-wuenschen-sich-fuenf-jahre-lang-smartphone-updates> (01.05.2020).

Wagner, E.; Poppe, E.; Hahn, F.; Jaeger-Erben, M.; Druschke, J.; Nissen, N. F.; Lang, K. –D. (2020): Decomposing Software Obsolescence Cases – A Cause and Effect Analysis Framework for Software Induced Product Replacement. In: International Congress "Electronics Goes Green 2020+", Proceedings, Berlin, ISBN: 978-3-8396-1659-8

Ward, C. D.; Sohns, C.W. (2011): Electronic component obsolescence. In IEEE Instrumentation & Measurement Magazine, vol. 14, no. 6, pp. 8-12, December 2011.

Watts, M. (2016): Software-driven technology insertion strategies for automatic test systems. 2016 IEEE AUTOTESTCON, Anaheim, CA, 2016, pp. 1-5.

Welch, C. (2019, December 30): Sonos explains why it bricks old devices with 'Recycle Mode'. An irreversible kill switch for a discount is a bad deal for the environment. In: The Verge Online Magazine. <https://www.theverge.com/2019/12/30/21042871/sonos-recycle-mode-trade-up-program-controversy> (12.02.2020).

Wernicke, S. (2019, May 07): Philips Hue: Support für Hue Bridge Version 1 endet. Das müsst ihr wissen. EURONICS Deutschland eG. <https://trendblog.euronics.de/smart-home/philips-hue-support-fuer-hue-bridge-version-1-endet-das-muesst-ihr-wissen-71857/> (06.08.2021).

Wirth, N., (1994): Gedanken zur Software-Explosion. Informatik Spektrum: Vol. 17, No. 1. Berlin Heidelberg: Springer-Verlag. (S. 5-10).

Wölbart, C. (2021, September 30): Fairphone 4: 5G, fünf Jahre Garantie und Versprechen für lange Update-Versorgung. <https://www.heise.de/news/Fairphone-4-5G-fuenf-Jahre-Garantie-und-Versprechen-fuer-lange-Update-Versorgung-6204938.html>.

Wu, Minghui; Hou, Honglun; Liu, Chang; Ying, Jing. (2006): COTS-based System's Obsolescence Risk Evaluation. Proceedings - 2006 10th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2006. 240-244. 10.1109/CSCWD.2006.253097.

Wubbena, H. (2007): Delivering on obsolescence protection, 2007 IEEE Autotestcon, Baltimore, MD, 2007, pp. 517-522.

Yeeply (2020): Mobile App Entwicklung: Android vs. iOS. Yeeply. <https://de.yeeply.com/blog/android-vs-ios-mobile-app-entwicklung/> (28.07.2021).

A Leitfaden Akteursinterviews



Leitfaden Interview mit Akteuren softwarebedingter Obsoleszenz

ISIconslt, 28.1.2021

Präambel (Vorab-Info):

Der Anteil an Geräten, der von Software abhängig ist, nimmt stetig zu. Haushaltsgeräte, Unterhaltungselektronik, Gebäudetechnik bis hin zu Fahrzeugen – sie alle werden heute von Software gesteuert. Immer häufiger entscheidet daher die Qualität der Software darüber, wie lange und wie gut ein „smartes“ Gerät genutzt werden kann.

Das Projekt „Analyse der software-basierten Einflussnahme auf eine verkürzte Nutzungsdauer von Produkten“ untersucht im Auftrag vom Umweltbundesamt systematisch die Ursachen, Ausprägungen und Zukunftstrends softwarebedingter Obsoleszenz, um auf dieser Grundlage politische Handlungsempfehlungen abzuleiten. Um die Ergebnisse mit Expert*innen zu diskutieren und zu validieren, führen wir derzeit Interviews mit verschiedenen Personen durch, die aufgrund ihrer Praxiserfahrungen, Verbandstätigkeiten, Unternehmenspraktiken oder Forschungsprojekte mit dem Phänomen softwarebedingte Obsoleszenz in Berührung kommen.

Die Interviews werden auf Band aufgenommen und inhaltlich ausgewertet. Die Aufnahmen werden gemäß den Regeln guten wissenschaftlichen Arbeitens nur intern verwendet. Ergebnisse und Zitate werden anonymisiert veröffentlicht.

TEIL I: Generelle Einstiegsfragen

1. Bitte erzählen Sie zunächst in welcher Form Sie mit Obsoleszenz, d.h. der frühzeitigen Alterung von Elektro(nik)geräten im Allgemeinen und softwarebedingter Obsoleszenz, d.h. der Alterung aufgrund von Software zu tun haben?
2. Was können aus Ihrer Sicht die Ursachen für softwarebedingte Obsoleszenz sein? Können Sie konkrete Beispiele nennen?
3. Mit der Zunahme an digitalen oder smarten Geräten in Alltag und Haushalten können in Zukunft immer mehr Fälle softwarebedingter Obsoleszenz auftreten. Hierbei sind verschiedene Akteure mit unterschiedlichen Einflussmöglichkeiten beteiligt.
 - a. Was können Produktentwickler*innen und Produkthersteller*innen tun, um softwarebedingte Obsoleszenz zu vermeiden? Welche Rolle spielen weitere Akteure der Wertschöpfung, wie Handel, Software-Anbieter oder Zulieferer?
 - b. Was können Nutzer*innen aus Ihrer Sicht tun, um softwarebedingte Obsoleszenz zu vermeiden?
 - c. Was sollte die Politik tun, um Unternehmen dabei zu unterstützen oder sie dazu zu bringen, Software bzw. das Ensemble von Software und Hardware bei smarten Geräten langlebiger zu gestalten?
 - d. Welche Akteure haben darüber hinaus die Möglichkeit, das Problem zu bearbeiten oder zu beeinflussen (z.B. Verbraucherschutz? Netzwerke oder Verbände wie Open Source-Initiativen oder kritische Anwender*innen)?



4. Sind Ihnen heute schon implizite oder explizite Vorgaben für die Mindestverfügbarkeit von Software in bestimmten Bereichen bekannt (Branchenstandards, freiwillige Vereinbarungen in Kaufverträgen etc.)?
5. Die Funktionalität und Nutzbarkeit vieler smarterer Geräte hängt von der Nutzung bestimmter Software ab. Was halten Sie von einer gesetzlich vorgeschriebenen Mindesthaltbarkeit von Software?
6. EU-weit soll das Recht auf Reparatur gefördert werden. Damit sind die Reparierfähigkeit von Produkten, die Verfügbarkeit von Reparaturdienstleistungen und die Weitergabe von Reparaturwissen gemeint.
 - a. Wie stehen Sie persönlich/ ihre Institution zum Thema Recht auf Reparatur?
 - b. Welche Rolle sollte Software aus Ihrer Sicht beim Thema Recht auf Reparatur spielen?
7. Welche neuen gesetzlichen Regelungen sind aus Ihrer Sicht notwendig oder welche bestehenden Gesetze oder Richtlinien müssen aus Ihrer Sicht angepasst werden, um softwarebedingte Obsoleszenz weitgehend zu verhindern?

TEIL II: Spezifische Fragen für Produktentwickler und -hersteller sowie Software-Anbieter

1. Wird das Thema softwarebedingte Obsoleszenz in ihrem Unternehmen/ ihrer Organisation thematisiert und bearbeitet? Wenn ja, wie?
2. Inwiefern spielen die Themen "Langlebigkeit" oder "Haltbarkeit" der Software bei der Entwicklung neuer Produkte (Software und Hardware) eine Rolle? Auch im Vergleich zu anderen Kriterien, wie Preis, Neuheitswert oder Ästhetik?
3. Werden konkrete Schritte unternommen oder Strategien gegen eine softwarebedingte Obsoleszenz entwickelt? Welche Zielkonflikte sehen Sie dabei in den Geschäftsmodellen?
4. Meist wird für smarte Geräte wie Smartphones eine Update-Garantie von drei Jahren gewährt. Was sind die Herausforderungen bei der Aufrechterhaltung solcher Update-Garantien?
5. Die Verfügbarkeit für Softwaresupport und Updates sind für die Nutzungsdauer von Smartphones entscheidend. Könnte aus Ihrer Sicht eine Update-Garantie über einen Zeitraum von 5 Jahren gewährt werden? Was spräche dafür, was dagegen?
6. Wie sehen die Entscheidungsprozesse in Ihrem Unternehmen/ihrer Organisation zur Abkündigung (End of Sales + Support) der Software aus?
 - a. Ursachen und Gründe



- b. Wer trifft die Entscheidungen? (interne Akteure, externe Stakeholder etc.)
 - c. Wann werden Entscheidungen getroffen?
 - d. Wie und wann werden Entscheidungen kommuniziert?
7. Nach Supportende für Geräte könnte Dritten Zugriff auf den Quellcode zur Weiterentwicklung verwendeter Software oder zur Weiterverwendung der Geräte mit externer freier Software gewährt werden. Was sind aus ihrer Sicht Potenziale und Hemmnisse einer solchen Praxis?
8. Inwieweit sind bestimmte proprietäre Produkte und Dienstleistungen sowie deren Software von einem erweiterten Recht auf Reparatur, wie es auf EU-Ebene vorgesehen ist, betroffen? Werden daraufhin Änderungen am Design von Produkten vorgenommen?

B Steckbriefe zu politischen Handlungsempfehlungen

B.1 Obsoleszenz durch externe Abhängigkeiten

Immer mehr softwarebetriebene Produkte sind vernetzt, also in der Lage, untereinander Informationen auszutauschen. Die vernetzten Produkte können entfernt (remote) oder mithilfe anderer Geräte synchronisiert und gesteuert oder in einen Cloudservice eingebunden werden. Dadurch entsteht eine große Vielzahl neuer Funktionalitäten und Dienstleistungen, die jedoch auch externe Abhängigkeiten bedingen und die Funktionalität des Gerätes für sich genommen einschränken.

B.1.1 Problem

Durch den hohen Vernetzungsgrad sind softwarebetriebene Produkte nicht mehr ein einzelnes Produkt, sondern ist in ein Produktsystem eingebettet, dessen Komponenten und Dienstleistungen von unterschiedlichen Herstellern oder Dienstleistern bereitgestellt werden. Insgesamt können Abhängigkeiten entstehen von:

1. einer Netzwerkanbindung (lokal oder Internet),
2. Cloud-Diensten, die Funktionalitäten für die Produktnutzung bereitstellen,
3. Vertriebsplattformen, die alte Softwareversionen verhindern, insbesondere App Stores, die als Gatekeeper auftreten.

Große Softwareanbieter vertreiben und verteilen Software-Applikationen und -Updates ausschließlich oder überwiegend über ihre eigene Plattform (z.B. Apple App Store, Google Playstore, Amazon Appstore), deren Zugang streng kontrolliert wird. Die Abhängigkeit zwischen den Plattformbetreiber und Kunden (bspw. Hersteller von smarten Leuchtmittel) sind sowohl technischer als auch organisatorischer und rechtlicher Art. Eine Garantiezusage des Herstellers für seine Produkte ist bei diesen Abhängigkeiten nur bedingt möglich. Diese Abhängigkeiten erhöhen die Wahrscheinlichkeit, dass softwarebetriebene Produkte vorzeitig ersetzt werden, weil die notwendige Software zum Betrieb der smarten und vernetzten Produkte nicht mehr kompatibel ist oder nicht mehr den Sicherheitsanforderungen entspricht. Die Folgen sind höherer Ressourcenbedarf und größeres Aufkommen von Elektroschrott auf eine vermeidbare Weise. Das gleiche Problem gilt für die Anhängigkeit zur Cloud-Plattform eines externen Anbieters. Wird der Cloud-Dienst abgekündigt oder stellt der Dienst ein Sicherheitsproblem dar, kann das Produkt nicht mehr genutzt werden, sofern es auf diesen Dienst angewiesen ist.

Beispiele:

1. Die Steuerung von Lampen in einem Smart Home System erfordert eine zentrale Steuerungseinheit, die nicht vom Hersteller der Lampe bereitgestellt wird. Der Hersteller der zentralen Steuerungseinheit entscheidet sich für ein neues (ggf. besseres) Übertragungsprotokoll, die Lampe ist allerdings nicht in der Lage, von der Steuerungseinheit das entsprechende Update zu erhalten.
2. Heimassistenten- und Sprachsteuerungssysteme werden zunehmend miteinander und mit anderen Diensten, wie z.B. Streaming-Diensten, vernetzt. Für einen Lautsprecher oder Fernseher sind dies externe Cloudservices, für deren Bereitstellung und Support der Hersteller in der Regel keine Garantieaussage treffen.

3. Apps werden entsprechend den Vorgaben von App Stores ausschließlich für das neueste Betriebssystem und die neueste Gerätehardware entwickelt, sodass die Installation vieler Apps auf funktionsfähigen älteren Geräten nicht mehr möglich ist. Nach einem Betriebssystemupdate ist die Funktionsfähigkeit anderer alter Apps jedoch nicht garantiert. Insgesamt laufen alte Apps nicht auf neuen Betriebssystemen und neue Apps nicht auf alten Betriebssystemen. Zudem werden alte Versionen des Betriebssystems vom Hersteller nach einer gewissen Zeit abgekündigt, sodass ein Zwang zu Updates von Betriebssystem und Apps entsteht. Gegebenenfalls unterstützt das neue Betriebssystem nicht mehr die alte Hardware.

Meist ist die durch die Abhängigkeiten entstehende Obsoleszenz technisch vermeidbar.

B.1.2 Lösungen

Technische, organisatorische und rechtliche Abhängigkeiten können reduziert oder umgangen werden. Folgende Ansätze zeigen die Machbarkeit einer Problemlösung.

Bei proprietären Übertragungsstandards und Protokollen muss eine Mindestsupportdauer des Produktsystems garantiert sein, die sicherstellt, dass auch beim Austausch von Komponenten des Produktsystems dieses in gleichbleibender Qualität funktioniert. Offene Standards reduzieren die Gefahr, dass nachteilige Abhängigkeiten entstehen. Unabhängige Vertriebswege von Software und Apps haben das Potenzial, längere Supportzeiten sowie Auf- und Abwärtskompatibilität zu ermöglichen. Offene Schnittstellen ermöglichen non-profit Organisationen, Software über den Zeitraum der Wirtschaftlichkeit hinaus zu pflegen.

Technische Abhängigkeiten, wie z.B. die dauerhafte Verfügbarkeit eines Clouddienstes oder einer Funkverbindung, können reduziert oder umgangen werden, indem Produkte im Produktdesign mit einer Fall-Back Option ausgestattet werden. Beispielsweise muss es möglich sein, die Kernfunktionalität von Geräten auch ohne eine Netzwerkanbindung zu bedienen. Zum Beispiel können Lampen einen Lichtschalter haben, Thermostate Knöpfe oder Drehregler und Lautsprecher können mit einem steckbaren Audioeingang ausgestattet sein etc.

Physisch intakte Geräte müssen nicht aufgrund ihrer Software zu Elektroschrott werden. Verkaufsplattformen von Apps und Gatekeeper sind oft mit den Hardwareherstellern oder deren Vertriebswegen wirtschaftlich und rechtlich verbunden (z.B. Apple, Google, Amazon). Diese Art von Geschäftsmodell sollte als unlauterer Wettbewerb gewertet werden. Der Missbrauch von Monopolstellungen kann, z.B. kartellrechtlich, adressiert werden. Eine sichere und transparentere Alternative wäre eine Pflicht zu Open-Source-Software.

Damit die Lösungen tatsächlich umgesetzt werden, müssen teilweise regulatorische Anreizstrukturen verändert werden. Diese sind im Folgenden genannt.

B.1.3 Regulatorische Ansatzpunkte

Ansatzpunkte, um technischen Abhängigkeiten regulatorisch zu adressieren, finden sich vor allem auf der europäischen Ebene in der Ökodesign-Richtlinie, der geplanten Initiative für nachhaltige Produkte und der Funkanlagenrichtlinie (radio equipment directive). Das Projektteam empfiehlt festzuschreiben, dass es bei softwarebetriebenen Produkten die

Möglichkeit zum Betrieb der Geräte ohne externe Abhängigkeiten geben muss. Um das regulatorische Korsett für Hersteller nicht zu eng zu schnüren, sollten Hersteller die Möglichkeit haben, die Kernfunktionalität selbst zu bestimmen. Damit Verbraucher*innen dennoch wissen, welche Erwartungen sie von dem Produkt haben dürfen, sollten Hersteller dazu verpflichtet werden, die Kernfunktionalität zu benennen. Die Aspekte des Verbraucher*innen-Schutzes betreffend, könnten beispielsweise in die Warenkaufrichtlinie Eingang finden.

Möglichkeit zum Betrieb der Geräte ohne externe Abhängigkeiten

- ▶ Softwarebetriebene Geräte sollten grundsätzlich ohne externe Abhängigkeiten betrieben werden können.
- ▶ Dies beinhaltet, dass die Geräte ihre Kernfunktionalität ohne Netzwerkanbindung („offline“) und ohne externe Cloud-Dienste erbringen müssen.
- ▶ Der volle Funktionsumfang darf durch die Entkopplung von externen Abhängigkeiten eingeschränkt sein (Fallback-Option).
- ▶ Die Hersteller müssen jedoch beim Verkauf der Produkte die Kernfunktionalitäten benennen, also welchen Funktionsumfang die Geräte im entkoppelten, unabhängigen Betriebsmodus noch erbringen.

Nennung der Abhängigkeiten von softwarebetriebenen Produkten

- ▶ Es sollte durch den Hersteller oder Händler des jeweiligen Produktes Transparenz darüber geschaffen werden, welche zusätzliche Hardware, Software oder Dienstleistung für den vollen Funktionsumfang des Produktes erforderlich sind (z.B. Notwendigkeit eines Smartphones unter Angabe des Betriebssystems und dessen Version, erforderliche Bandbreite im Netzwerk).
- ▶ Pflicht zur Nennung der erforderlichen externen Dienstleistungen oder Cloud-Services, die zur Erfüllung des vollen Funktionsumfangs benötigt werden (z.B. Benutzerkonto bei einem Cloud-Service).
- ▶ Pflicht zur Nennung von Art und Umfang der über Netzwerke versendeten Daten, Zweck der externen Verarbeitung und Angaben zum Datenschutz.

Anforderungen an externe Dienstleistungen

- ▶ Sind zum Betrieb eines softwarebetriebenen Produktes externe Dienstleistungen erforderlich, beispielsweise eine externe Cloud-Dienstleistung, so muss der Hersteller eines derart abhängigen Produktes durch vertragliche Vereinbarungen mit dem externen Dienstleister sicherstellen, dass die externe Dienstleistung für mindestens 10 Jahre aufrechterhalten wird.
- ▶ Diese Anforderung kann entfallen, wenn das Produkt auch ohne externe Abhängigkeiten in gleichbleibender Qualität betrieben werden kann.

Anforderungen an Verkaufs- und Vertriebsplattformen von Software/Apps

- ▶ Verkaufs- und Vertriebsplattformen von Software/Apps dürfen ihre marktbeherrschende Stellung nicht dazu missbrauchen, softwarebedingte Obsoleszenz herbei zu führen.
- ▶ So sollten Verkaufs- und Vertriebsplattformen, die eine vorherrschende Marktstellung aufweisen (z.B. Marktanteil von mehr als 50% für bestimmte Produkte), durch ihre Vertragsbedingungen keine Apps von dem Vertrieb über ihre Plattform ausschließen, die für ältere Hardware und für ältere Betriebssysteme programmiert sind. Dadurch soll sichergestellt werden, dass auch ältere Geräte mit neuer Software bzw. Software-Updates versorgt werden können.
- ▶ Entwicklern von open-source Software muss es kostenfrei gestattet und ermöglicht werden, nicht-kommerzielle Software und Apps für alle Betriebssysteme und Versionen zur Verfügung zu stellen, sodass diese Software von Konsument*innen kostenfrei installiert und genutzt werden kann.

B.2 Obsoleszenz bei Systemprodukten

Softwarebetriebene und vernetzte Geräte werden als „Systemprodukt“ bezeichnet, wenn ein Hersteller die Funktionalität eines Produktes auf verschiedene Produkte aufteilt, die physisch trennbar und ggf. separat zu erwerben sind. Eine zentrale Steuerungseinheit verwaltet dabei verschiedene Funktionalitäten.

B.2.1 Problem

Ein erhöhter Ressourcenbedarf entsteht, wenn die starke Vernetzung dazu genutzt wird, eine Funktionalität, die durch ein einziges nicht vernetztes Gerät erbracht werden könnte, auf mehrere Geräte aufgeteilt wird, die alle notwendig sind, um dieselbe Funktionalität zu erbringen, ohne einen großen Mehrwert zu bieten. Zudem wird dieser modulare Ansatz oft dermaßen ausgestaltet, dass sich die Komplexität und Fehleranfälligkeit des Gesamtsystems erhöht. Der Ausfall einer Komponente des Produktsystems kann zum Versagen des gesamten Produktsystems führen. Stattdessen sollte Modularität genutzt werden, um die Reparierbarkeit zu verbessern und eine individuelle Anpassung des Funktionsumfangs zu ermöglichen. Die einzelnen Sensoren und Funkchips sind aber für Konsument*innen nicht austauschbar oder reparierbar und Übertragungswege, -protokolle und -standards sind oftmals nicht so publiziert und dokumentiert, dass sie von Dritten für eine Funktionalitätserweiterung genutzt werden können.

Beispiele:

1. Eine Produktfamilie aus (mehreren) vernetzten Lampen, Schaltern und einer Steuerungseinheit ist darauf angewiesen, dass alle Komponenten funktionsfähig sind, um das Licht einzuschalten. Ist einer der Funkchips nicht erreichbar, stellt der Hersteller einen Übertragungsweg ein, wird ein Protokoll abgekündigt oder ist dieses aus Gründen der Datensicherheit nicht mehr nutzbar, dann kann die Lampe nicht mehr ein- und ausgeschaltet werden. Sie wird obsolet, obwohl sie technisch noch dazu in der Lage wäre, zu leuchten.
2. Die Steuerung eines Heizkörperthermostaten wird als Cloudservice angeboten, sodass globale Wetterdaten und lokale Gegebenheiten (bspw. Nutzungsverhalten) miteinbezogen werden. Wird der Cloudservice eingestellt, entfällt die Hauptfunktionalität des Produktes und das Produkt wird obsolet. Die Obsoleszenz wird dadurch begünstigt, dass die Schnittstellen und Übertragungsprotokolle des vernetzten Produktes nicht öffentlich dokumentiert sind und dadurch keine alternativen Steuermöglichkeiten entwickelt werden können. Modularität sollte stattdessen ermöglichen, verschiedene Cloudservices mit demselben Gerät zu nutzen und dadurch keine einseitigen Abhängigkeiten schaffen.

B.2.2 Lösungen

Da die gegenseitigen Abhängigkeiten innerhalb einer Produktfamilie bestehen, ist der Hersteller der Produkte gleichzeitig Dienstleister des Softwaresystems. Der Hersteller wäre technisch in der Lage, sicherzustellen, dass einzelne Komponenten und deren Schnittstellen nicht zur Obsoleszenz des Gesamtsystems führen. Dies sollte er in seinem Produktdesign berücksichtigen.

Offene Schnittstellen würde es zusätzlich anderen Herstellern oder open-source Gemeinschaften ermöglichen, den Support zu übernehmen, da insbesondere lange Supportzeiten für Software ein finanzielles Risiko darstellen. Die Hersteller sind insbesondere in der Lage, für ihre Produktfamilie eine Mindestnutzungsdauer inklusive funktionaler und sicherheitsrelevanter Updates zu garantieren. Dazu müssten Hersteller verpflichtet werden, Komponenten und Teilprodukte des Produktsystems über die gesamte Mindestnutzungsdauer als Ersatzteile bereit zu halten, entsprechende Updates bereit zu stellen und Funktionsgarantien zu geben. Eine Öffnung von Systemprodukten für weitere Marktteilnehmer während der wirtschaftlichen Phase des Produktsystems verbreitert die Ersatzteilverfügbarkeit und Kompatibilität darüber hinaus.

B.2.3 Regulatorische Ansatzpunkte

Ansatzpunkte, um Obsoleszenz durch organisationsinterne Abhängigkeiten regulatorisch zu adressieren, finden sich beispielsweise in der Warenkaufrichtlinie und der Ökodesign-Richtlinie und speziell für Leuchtmittel in der Verordnung (EU) 2019/2020 zur Festlegung von Ökodesign-Anforderungen an Lichtquellen und separate Betriebsgeräte.

Nennung der Abhängigkeiten von softwarebetriebenen Produkten

- ▶ Es sollte durch den Hersteller oder Händler des jeweiligen Produktes Transparenz darüber geschaffen werden, welche zusätzliche Hardware, Software oder Dienstleistung für den vollen Funktionsumfang des Produktes erforderlich sind (z.B. Notwendigkeit eines Smartphones unter Angabe des Betriebssystems und dessen Version, erforderliche Bandbreite im Netzwerk).
- ▶ Pflicht zur Nennung der erforderlichen externen Dienstleistungen oder Cloud-Services, die zur Erfüllung des vollen Funktionsumfangs benötigt werden (z.B. Benutzerkonto bei einem Cloud-Service).
- ▶ Pflicht zur Nennung von Art und Umfang der über Netzwerke versendeten Daten, Zweck der externen Verarbeitung und Angaben zum Datenschutz.

Verpflichtende Angabe des garantierten Supportzeitraums und daraus resultierende Rechtsansprüche

- ▶ Hersteller von softwarebetriebenen Produkten sollten beim Verkauf von Produkten und vernetzten Produktsystemen verpflichtet werden, den garantierten Supportzeitraum für Software-Updates durch die Nennung des Support-Enddatums anzugeben und zu beschreiben, worauf sich dieser Support bezieht.
- ▶ Während des genannten Zeitraums vom Kauf bis zum Support-Enddatum haben die Käufer*innen des Produktes einen Rechtsanspruch auf die Bereitstellung von Software-Updates und der genannten Unterstützungen. Dieser Anspruch gilt auch für gebraucht gekaufte Geräte.
- ▶ Die Verpflichtung zur Deklaration des garantierten Supportzeitraums kann bereits eingeführt werden, bevor es einen Mindestzeitraum für den Software-Support oder eine Mindestnutzungsdauer gibt (siehe Punkt 3: Gewährleistung einer Mindestnutzungsdauer).

- ▶ Für den Fall, dass der Hersteller keinen Support über die gesetzliche Gewährleistungsfrist hinaus anbietet, muss dies ebenfalls deklariert werden (z.B. „kein Support“).

Gewährleistung einer Mindestnutzungsdauer

- ▶ Der Hersteller von softwarebetriebenen Produkten muss gewährleisten, dass seine Produkte über einen Mindestzeitraum in gleichbleibender Qualität nutzbar sind und muss die entsprechenden Voraussetzungen dafür schaffen.
- ▶ Innerhalb der Mindestnutzungsdauer besteht für den Hersteller die Pflicht zu Software-Updates für den Erhalt aller Funktionen, die zum Übergabezeitpunkt zur Verfügung standen. Software-Updates dürfen die Ausführungsgeschwindigkeit der Funktionen des Produktes nicht mehr als 30% reduzieren und den Energiebedarf insgesamt nicht mehr als 10% erhöhen.
- ▶ Zur Gewährleistung der Mindestnutzungsdauer gehören insbesondere die vertraglichen Vereinbarungen mit eigenen Lieferanten und externen Dienstleistern und mit Herstellern von Teilprodukten eines zusammenhängenden Produktsystems. Dies sind beispielsweise die Ersatzteilbereitstellung (z.B. Akkus, Speichermedien, abnutzende Bauteile, Steckernetzteile), Bereitstellung kompatibler Produkte eines Produktsystems (z.B. Datenbrücken/Bridges, zentrale Steuereinheiten, verteilte Sensoren und Aktoren oder Fernbedienungen), Bereitstellung von Software-Updates (z.B. für die Bedienbarkeit des Produkts selbst und die sichere Datenübertragung), Aufrechterhaltung externer Cloud-Dienste (z.B. zur Datenlieferung, -verarbeitung und -speicherung). Vertragspartnerschaften zusammengehöriger Dienste und Geräte von Produktsystemen müssen entsprechend ausgestaltet werden.
- ▶ Diese Mindestnutzungsdauer sollte deutlich über die gesetzliche Gewährleistungsfrist von 2 Jahren hinaus gehen und sich an der technischen Lebensdauer der Hardware orientieren. Auch bei Geräten, die unerwartet lange physisch einwandfrei sind, sollte Software nicht zur Obsoleszenz führen.
- ▶ Es wird eine Mindestnutzungsdauer von regelmäßig 10 Jahren vorgeschlagen, die nur in begründeten Ausnahmefällen kürzer sein darf.

Anforderungen an die Kompatibilität und Interoperabilität von Produktsystemen

- ▶ Der Hersteller von Produkten, die zu einem Produktsystem gehören (z.B. einzelne Steuereinheiten, Sensoren und Aktoren einer Smart-Home-Anwendung), müssen dafür Sorge tragen, dass ihr Produkt über die Mindestnutzungsdauer mit dem zugehörigen Produktsystem zusammen genutzt werden kann, auch dann, wenn sich das Produktsystems weiterentwickelt.
- ▶ Dies muss er entweder durch die Aufrechterhaltung der Kompatibilität seines Produktes zu dem zugehörigen Produktsystem (beispielsweise durch Software-Updates am Produkt selbst) oder durch vertragliche Vereinbarungen mit dem Hersteller des Produktsystems erreichen (z.B. durch Vereinbarung der Rückwärtskompatibilität des Produktsystems zu früheren Produkten).
- ▶ Die Pflicht zu vertraglichen Vereinbarungen mit Dritten entfällt, wenn im Produkt allgemein anerkannte und offen zugängliche Standards verwendet werden, die auch mit anderen

Produktsystemen kompatibel sind (z.B. genormte physische Schnittstellen und offene Übertragungsprotokolle).

- ▶ Die Hersteller von Produktsystemen müssen ihrerseits zur Kompatibilität beitragen, indem sie anderen Marktteilnehmern die Interoperabilität mit ihrem Produktsystem ermöglichen, z.B. indem die Schnittstellen und Übertragungsprotokolle des Produktsystems offenen Standards entsprechen und öffentlich zugänglich dokumentiert werden.

Bereitstellung von sicherheitsrelevanten Software-Updates über einen Mindestzeitraum

- ▶ Hersteller von Produkten sollten dazu verpflichtet werden, über einen Mindestzeitraum Software-Updates bereit zu stellen, die einen sicheren Betrieb der Produkte zulassen (Schutz der Gesundheit, der Privatsphäre und vor finanziellem Verlust).
- ▶ Der Zeitraum beginnt mit dem Ende des Inverkehrbringens des letzten Exemplars eines Modells der jeweiligen Produkte und sollte mindestens 10 Jahre betragen.

Kennzeichnung von Produkten zu ihrer Reparierbarkeit

- ▶ Softwarebetriebene Produkte sollen europaweit mit einer leicht verständlichen Kennzeichnung über ihre Reparierbarkeit versehen werden, nach der Art des französischen Reparatur-Scores (vgl. Kapitel 8.2.5).
- ▶ Die garantierte Supportdauer und die Wartbarkeit der im Produkt enthaltenen Software sollten zusätzlich im Reparatur-Score berücksichtigt werden. Wartbare Software kann z.B. ohne Einschränkungen sowohl up-graded (in eine höhere Version versetzt) als auch down-graded (in eine niedrigere Version versetzt) werden und ist modular aufgebaut.

B.3 Direkte softwarebedingte Obsoleszenz

Software ist ein immaterielles Gut und verhält sich anders als physische Produkte. Sie wird meist unvollständig, mit Grundfunktionen ausgestattet, veröffentlicht und „reift nach“, d.h. sie wird an Erwartungen und sich verändernde Gegebenheiten angepasst. Jedoch kann Software, wenn sie nicht weiterentwickelt wird, relativ zu ihrer Umgebung altern.

B.3.1 Problem

Sowohl die nachlassende Aktualität als auch der wachsende Speicher- und Hardwarebedarf von Software nach dessen Aktualisierung führen zu direkter Obsoleszenz von Hardware. Alte Software birgt meist Sicherheitsrisiken, da sich auch Entschlüsselungs- und Hackertechnologien ständig weiterentwickeln. Updates stellen meistens jedoch Funktionserweiterungen oder ein neues Design bereit, die einen erhöhten Bedarf an Rechenleistung, Speicherplatz und Permanentspeicher zur Folge haben. Ein starkes Wachstum der Ansprüche einer Software wird als „*Softwarebloat*“ bezeichnet.

Des Weiteren wird Software dazu genutzt, Reparatur und Betrieb einzuschränken oder zu verhindern. Hersteller argumentieren in diesen Fällen, dass solche Maßnahmen dem Erhalt der Qualität des Produktes dienen würden. In manchen Fällen verlängern die Maßnahme tatsächlich die Lebensdauer des Produktes, rechtlich sind solche Eingriffe des Herstellers mit Risiken verbunden, wenn die Hersteller ihre Kunden nicht ausreichend darüber informieren.

Beispielsweise

1. Ein Smartphone birgt Sicherheitsrisiken, wenn es keine Sicherheitsupdates mehr erhält. Der Anbieter des Betriebssystems hat zwar entsprechende Software veröffentlicht, der Hardware-Hersteller bekommt jedoch von seinem Zulieferer kein Update für die Treiber der verbauten Chips.
2. Ein Tablet wird mit jedem Update langsamer, weil das Betriebssystem mit jedem Update mehr Hardwareressourcen benötigt.
3. Ein Smartphone, dessen Display einen Sprung hat, kann nicht repariert werden, da es das baugleiche Ersatzteil des nicht-autorisierten Fachhändlers erkennt und mit einer Fehlermeldung ablehnt. Dies wird als „*parts pairing*“ bezeichnet.
4. Ein Drucker stellt softwareseitig seinen Betrieb ein, da die Zählvorrichtung gedruckter Seiten eine Schwelle überschritten hat.

B.3.2 Lösungen

Software darf nicht dazu führen, dass physisch intakte Geräte vorzeitig zu Elektroschrott werden. Im Gegenteil hat sie sogar das Potential, sich verlängernd auf die Lebensdauer auszuwirken.

Sollten sich die Rahmenbedingungen einer sicheren Nutzung stark ändern (z.B. durch neue Malware oder drohende Akkuexplosion), kann die Software durch eine Fall-back-Option auf ihre Kernfunktionalität beschränkt werden. Im Voraus ist darüber Transparenz zu schaffen, welche Funktionalitäten wie eingeschränkt werden können.

B.3.3 Regulatorisch Ansatzpunkte

Ansatzpunkte, um softwarebedingte Obsoleszenz zu adressieren, finden sich insbesondere im Verbraucherschutz, z.B. der Warenkaufrichtlinie und der Digitale Inhalte Richtlinie. Software als Verhinderer von Reparaturen könnte über die Ökodesign-Richtlinie adressiert werden. Die Beschaffung umweltfreundlicher Produkte sollte in der Allgemeine Verwaltungsvorschrift zur Beschaffung klimafreundlicher Leistungen ergänzt werden.

Verpflichtende Angabe des garantierten Supportzeitraums und daraus resultierende Rechtsansprüche

- ▶ Hersteller von softwarebetriebenen Produkten sollten beim Verkauf von Produkten und vernetzen Produktsystemen verpflichtet werden, den garantierten Supportzeitraum für Software-Updates durch die Nennung des Support-Enddatums anzugeben und zu beschreiben, worauf sich dieser Support bezieht.
- ▶ Während des genannten Zeitraums vom Kauf bis zum Support-Enddatum haben die Käufer*innen des Produktes einen Rechtsanspruch auf die Bereitstellung von Software-Updates und der genannten Unterstützungen. Dieser Anspruch gilt auch für gebraucht gekaufte Geräte.
- ▶ Die Verpflichtung zur Deklaration des garantierten Supportzeitraums kann bereits eingeführt werden, bevor es einen Mindestzeitraum für den Software-Support oder eine Mindestnutzungsdauer gibt (siehe Punkt 3: Gewährleistung einer Mindestnutzungsdauer).
- ▶ Für den Fall, dass der Hersteller keinen Support über die gesetzliche Gewährleistungsfrist hinaus anbietet, muss dies ebenfalls deklariert werden (z.B. „kein Support“).

Bereitstellung von sicherheitsrelevanten Software-Updates über einen Mindestzeitraum

- ▶ Hersteller von Produkten sollten dazu verpflichtet werden, über einen Mindestzeitraum Software-Updates bereit zu stellen, die einen sicheren Betrieb der Produkte zulassen (Schutz der Gesundheit, der Privatsphäre und vor finanziellem Verlust).
- ▶ Der Zeitraum beginnt mit dem Ende des Inverkehrbringens des letzten Exemplars eines Modells der jeweiligen Produkte und sollte mindestens 10 Jahre betragen.

Gewährleistung einer Mindestnutzungsdauer

- ▶ Der Hersteller von softwarebetriebenen Produkten muss gewährleisten, dass seine Produkte über einen Mindestzeitraum in gleichbleibender Qualität nutzbar sind und muss die entsprechenden Voraussetzungen dafür schaffen.
- ▶ Innerhalb der Mindestnutzungsdauer besteht für den Hersteller die Pflicht zu Software-Updates für den Erhalt aller Funktionen, die zum Übergabezeitpunkt zur Verfügung standen. Software-Updates dürfen die Ausführungsgeschwindigkeit der Funktionen des Produktes nicht mehr als 30% reduzieren und den Energiebedarf insgesamt nicht mehr als 10% erhöhen.

- ▶ Zur Gewährleistung der Mindestnutzungsdauer gehören insbesondere die vertraglichen Vereinbarungen mit eigenen Lieferanten und externen Dienstleistern und mit Herstellern von Teilprodukten eines zusammenhängenden Produktsystems. Dies sind beispielsweise die Ersatzteilbereitstellung (z.B. Akkus, Speichermedien, abnutzende Bauteile, Steckernetzteile), Bereitstellung kompatibler Produkte eines Produktsystems (z.B. Datenbrücken/Bridges, zentrale Steuereinheiten, verteilte Sensoren und Aktoren oder Fernbedienungen), Bereitstellung von Software-Updates (z.B. für die Bedienbarkeit des Produkts selbst und die sichere Datenübertragung), Aufrechterhaltung externer Cloud-Dienste (z.B. zur Datenlieferung, -verarbeitung und -speicherung). Vertragspartnerschaften zusammengehöriger Dienste und Geräte von Produktsystemen müssen entsprechend ausgestaltet werden.
- ▶ Diese Mindestnutzungsdauer sollte deutlich über die gesetzliche Gewährleistungsfrist von 2 Jahren hinaus gehen und sich an der technischen Lebensdauer der Hardware orientieren. Auch bei Geräten, die unerwartet lange physisch einwandfrei sind, sollte Software nicht zur Obsoleszenz führen.
- ▶ Es wird eine Mindestnutzungsdauer von regelmäßig 10 Jahren vorgeschlagen, die nur in begründeten Ausnahmefällen kürzer sein darf.

Verbot von softwareseitigem Verhindern von Reparatur und Betrieb

- ▶ Die in Produkten integrierte Software darf nicht dazu genutzt werden, die Reparatur und den Betrieb von Produkten zu verhindern, indem beispielsweise Ersatzteile oder Betriebsmittel anderer Hersteller ausgeschlossen werden („parts pairing“).
- ▶ Sofern die Software für die Betriebssicherheit der Geräte zuständig ist (z.B. Überladungsschutz für Akkus) müssen die Standards offengelegt werden, die Reparaturbetriebe und Ersatzteilhersteller in die Lage versetzen, ebenso sichere Produkte einzubauen.
- ▶ Die Software darf ebenfalls nicht dazu genutzt werden, die Nutzungsdauer von Produkten zu begrenzen (z.B. Countdown- oder Abschaltvorrichtungen) oder deren Qualität und Funktionalität zu reduzieren.

Anforderungen an die Kompatibilität und Interoperabilität von Produktsystemen

- ▶ Der Hersteller von Produkten, die zu einem Produktsystem gehören (z.B. einzelne Steuereinheiten, Sensoren und Aktoren einer Smart-Home-Anwendung), müssen dafür Sorge tragen, dass ihr Produkt über die Mindestnutzungsdauer mit dem zugehörigen Produktsystem zusammen genutzt werden kann, auch dann, wenn sich das Produktsystems weiterentwickelt.
- ▶ Dies muss er entweder durch die Aufrechthaltung der Kompatibilität seines Produktes zu dem zugehörigen Produktsystem (beispielsweise durch Software-Updates am Produkt selbst) oder durch vertragliche Vereinbarungen mit dem Hersteller des Produktsystems erreichen (z.B. durch Vereinbarung der Rückwärtskompatibilität des Produktsystems zu früheren Produkten).
- ▶ Die Pflicht zu vertraglichen Vereinbarungen mit Dritten entfällt, wenn im Produkt allgemein anerkannte und offen zugängliche Standards verwendet werden, die auch mit anderen

Produktsystemen kompatibel sind (z.B. genormte physische Schnittstellen und offene Übertragungsprotokolle).

- ▶ Die Hersteller von Produktsystemen müssen ihrerseits zur Kompatibilität beitragen, indem sie anderen Marktteilnehmern die Interoperabilität mit ihrem Produktsystem ermöglichen, z.B. indem die Schnittstellen und Übertragungsprotokolle des Produktsystems offenen Standards entsprechen und öffentlich zugänglich dokumentiert werden.

Kennzeichnung von Produkten zu ihrer Reparierbarkeit

- ▶ Softwarebetriebene Produkte sollen europaweit mit einer leicht verständlichen Kennzeichnung über ihre Reparierbarkeit versehen werden, nach der Art des französischen Reparatur-Scores (vgl. Kapitel 8.2.5).
- ▶ Die garantierte Supportdauer und die Wartbarkeit der im Produkt enthaltenen Software sollten zusätzlich im Reparatur-Score berücksichtigt werden. Wartbare Software kann z.B. ohne Einschränkungen sowohl up-graded (in eine höhere Version versetzt) als auch down-graded (in eine niedrigere Version versetzt) werden und ist modular aufgebaut.

Verpflichtende umweltverträgliche öffentliche Beschaffung

- ▶ Die öffentliche Beschaffung von umweltverträglichen Produkten ist derzeit weitestgehend optional oder steht in Konkurrenz zu anderen Anforderungen wie z.B. der Wirtschaftlichkeit. In Anbetracht der Dringlichkeit von Klima- und Ressourcenschutz zur Erhaltung unserer Lebensgrundlage, sollte die öffentliche Beschaffung von umweltverträglichen Produkten verpflichtend werden.
- ▶ Dies sollte an zentraler Stelle mit höchster Priorität, beispielsweise in der Bundeshaushaltsordnung und den jeweiligen Länderverordnungen festgeschrieben werden.
- ▶ Zur Vermeidung von softwarebedingter Obsoleszenz sollten von öffentlichen Einrichtungen nur solche Produkte und Produktsysteme beschafft werden dürfen, die Mindestnutzungsdauern (siehe Punkt 3: Gewährleistung einer Mindestnutzungsdauer) gewährleisten können.
- ▶ Die Pflicht zur umweltverträglichen öffentlichen Beschaffung trifft insbesondere auf die Beschaffung von Informationstechnik im Rahmen des „Digitalpakts“ zur Ausstattung von Bildungseinrichtungen mit Computern und Tablets, aber auch für die Digitalisierung von öffentlichen Verwaltungen insgesamt und die dazu verwendeten Haushaltsmittel zu. Die Verwendung dieser Digitalisierungsmittel sollte grundsätzlich an Umwelt- und Langlebigkeitsanforderungen an die Produkte gekoppelt werden.
- ▶ Softwarebetriebene Geräte, für die keine ambitionierte Mindestnutzungsdauer gewährleistet wird, sollten unter den nicht mehr zu beschaffenden Produkten in der Allgemeine Verwaltungsvorschrift der Bundesregierung zur Beschaffung klimafreundlicher Leistungen (AVV Klima) in Anlage 1 gelistet werden.

Förderung und Lehre nachhaltiger Softwareentwicklung

- ▶ Der große Vorteil von Software ist, dass sie nicht zwangsläufig an gerätespezifische Beschränkung gebunden ist. Ganz im Gegenteil, anders als fest verdrahtete Logikbausteine, kann sie Maschinen universell und an neue Aufgabenstellungen anpassbar machen („Turingmaschine“). Software kann daher insbesondere auch zur Verlängerung von Produktnutzungsdauern beitragen.
- ▶ Um dies wieder stärker in die Praxis zu bringen, sollte die Entwicklung von nachhaltiger Software und Software, die die Produktlebensdauer verlängert, gefördert werden. Dies kann auf der Ebene von Hochschulen erfolgen (Integration in die Lehre), auf der Ebene von öffentlicher Auftragsvergabe (Entwicklung ressourcenschonender Anwendungen) sowie auf europäischer Ebene bei der Schaffung einer europäischen Cloud-Infrastruktur (GAIA-X).
- ▶ Öffentlich geförderte Software sollte grundsätzlich frei zugänglich und quelloffen sein, damit sie eine möglichst breite Wirkung entfalten kann.

C Anhang: Fragebogen für repräsentative Nutzer*innen-Umfrage

Im Folgenden stellen wir einige Fragen zu den Geräten, die Sie regelmäßig nutzen und Ihren Erfahrungen damit. Im Zentrum stehen dabei vor allem smarte Geräte, d.h. die mit dem Internet oder anderen Geräten vernetzt sind. Bitte beantworten Sie die Fragen nach bestem Wissen und Gewissen und nehmen Sie sich gerne Zeit, wenn Sie etwas nachdenken müssen.

Teil 1: Geräte-Besitz und Nutzungspraktiken

Wir nennen im Folgenden einige digital gesteuerte und vernetzte Geräte. Seit wann nutzen Sie welche der folgenden Geräte wie häufig in Ihrem eigenen Haushalt?

- SmartTV (Fernsehgerät mit Internetanschluss und Apps)
- Digital steuerbare Steckdosen
- Digital steuerbare Beleuchtung
- Digital steuerbare Heizung
- Digital steuerbare Waschmaschine
- Digital steuerbarer Kühlschrank
- Saugroboter
- Digitale Sicherheitsanlagen
- Digitale Türschlösser
- Kameras mit Internetanschluss
- Digital steuerbare Fenster
- Smart Speaker (Sprachsteuerungen wie Alexa, Google Assistant, etc.)
- Smart Watch
- Smartphone
- Internet-Router
- Spielkonsolen (mit virtual reality headsets und Streaming Funktionen)
- Set Top Box
- Notebook
- Tablet
- Vernetzter Drucker
- Router

Teil 2: Umgangsweisen und Erfahrungen mit Software, Funktionalität und Leistungsfähigkeit der Geräte

Bei welchem der oben genannten Geräte haben Sie mit der Zeit eine Verschlechterung, zum Beispiel in der Leistung oder Bedienbarkeit festgestellt?

- SmartTV (Fernsehgerät mit Internetanschluss und Apps)
- Digital steuerbare Steckdosen
- Digital steuerbare Beleuchtung
- Digital steuerbare Heizung
- Digital steuerbare Waschmaschine
- Digital steuerbarer Kühlschrank
- Saugroboter
- Digitale Sicherheitsanlagen

- Digitale Türschlösser
- Kameras mit Internetanschluss
- Digital steuerbare Fenster
- Smart Speaker (Sprachsteuerungen wie Alexa, Google Assistent, etc.)
- Smart Watch
- Smartphone
- Internet-Router
- Spielkonsolen (mit virtual reality headsets, Streaming-Funktionen)
- Set Top Box
- Notebook
- Tablet
- Vernetzter Drucker
- Router

2.2 Welche der folgenden Verschlechterungen haben Sie festgestellt? Mehrfach-Antworten sind möglich. [--> Filter: alle Geräte mit starker oder sehr starker Verschlechterung]

2.2.1 „Gerät a“

- Gerät verbraucht mehr Energie als zu Beginn.
- Gerät bringt nicht mehr die volle Leistung.
- Gerät funktioniert nicht mehr in vollem Umfang bzw. fallen regelmäßig Funktionen aus.
- Gerät meldet häufig Fehler.
- Gerät lässt sich schlechter bedienen als zu Beginn, z.B. reagiert es nur langsam auf Eingaben.
- Die Sicherheit ist eingeschränkt, weil keine Updates mehr möglich sind.
- Wegen ausbleibender Updates fallen Funktionen und Apps aus.
- Gerät kann sich nicht mehr oder nur noch schlecht mit anderen Geräten vernetzen.
- Das Gerät ist nicht mehr kompatibel mit neuen Apps, Standards oder Medienformaten, d.h. neue Programme oder Medien lassen sich nicht abspielen.
- Das Gerät kann keine Verbindung (mehr) zum Internet herstellen.
- Das Gerät muss häufig neu gestartet werden, um sich wieder mit anderen Geräten zu verbinden.
- Der Speicherplatz reicht nicht mehr aus.
- Das Gerät wird immer langsamer.
- Weitere Gründe: [offenes Textfeld, 500 Zeichen]

2.3 Wie gehen Sie aktuell mit den Problemen um? Bitte geben Sie Ihre Zustimmung zu folgenden Aussagen an. Mehrfachantworten sind möglich. [--> Filter 2.1: Alle Geräte mit „kaum verschlechtert, etwas verschlechtert, stark verschlechtert, Sehr stark verschlechtert“]; 5-stufige Skala: Stimme gar nicht zu, Stimme eher nicht zu, unbestimmt, Stimme eher zu, Stimme voll und ganz zu)

- Ich nutze das Gerät trotz seiner Einschränkungen.
- Ich nutze das Gerät nicht mehr. Es liegt ungenutzt zu Hause rum.
- Ich nutze das Gerät nicht mehr, aber andere nutzen es.

- Ich werde das Gerät entsorgen und mir ein Neues beschaffen.
- Ich hole mir Hilfe aus meinem Bekannten- und Familienkreis.
- Ich verkaufe das Gerät über einschlägige Marktplätze im Internet.
- Ich informiere mich in Internetforen und auf Hilfeseiten über den richtigen Umgang mit dem Problem.
- Ich hole mir Hilfe bei Reparaturdienstleistern.
- Ich versuche selbst, das Gerät zu reparieren.
- Ich beschwere mich auf Kundenportalen oder ähnlichen Plattformen.
- Sonstige [offenes Textfeld, 500 Zeichen]

Teil 3: Konsum-Bedürfnisse und Konsument*innen-Souveränität

3.1 Wenn Sie eines der folgend genannten vernetzten Geräte nicht besitzen: Wie wahrscheinlich ist es, dass Sie sich eines der folgenden Geräte in der nächsten Zeit anschaffen?

[--> Filter 1.1 B. „Nutze ich nicht“, 5-stufige Skala: Ganz und gar nicht, Eher unwahrscheinlich, unbestimmt, Eher wahrscheinlich, Sehr wahrscheinlich]

- SmartTV (Fernsehgerät mit Internetanschluss und Apps)
- Digital steuerbare Steckdosen
- Digital steuerbare Beleuchtung
- Digital steuerbare Heizung
- Digital steuerbare Waschmaschine
- Digital steuerbarer Kühlschrank
- Saugroboter
- Digitale Sicherheitsanlagen
- Digitale Türschlösser
- Kameras mit Internetanschluss
- Digital steuerbare Fenster
- Smart Speaker (Sprachsteuerungen wie Alexa, Google Assistent, etc.)
- Smart Watch
- Smartphone
- Internet-Router
- Spielkonsolen mit virtual reality headsets, Streaming Funktionen
- Set Top Box
- Notebook
- Tablet
- Vernetzter Drucker
- Router

3.2 Aus welchen Gründen planen Sie die Anschaffung? Wie sehr stimmen Sie den folgenden Aussagen zu: [--> Filter 3.1: Alle Geräte mit eher wahrscheinlich, Sehr wahrscheinlich]; 5-stufige Skala mit: Stimme gar nicht zu; Stimme eher nicht zu; unbestimmt; Stimme eher zu; Stimme voll und ganz zu]

- Ich möchte technisch auf dem neuesten Stand sein.
- Ich erhoffe mir eine Erleichterung des Alltags.
- Ich möchte ein neues Gerät, weil mich die neuen Funktionen begeistern.
- Ich brauche das Gerät, um andere Geräte, die ich bereits besitze, richtig nutzen zu können.

- Ich möchte etwas Neues ausprobieren.
- Ich habe ein günstiges Angebot gesehen.
- Mir wurde das Gerät von jemandem empfohlen.
- Ich habe das Gerät bei anderen gesehen und möchte es auch haben.
- Digitale Geräte sind für mich ein Muss.
- Ich möchte ein neues Gerät, bevor mein altes kaputtgeht.
- Ich möchte ein neues Gerät, weil mein altes Gerät Mängel hat.
- Mit dem neuen Gerät leiste ich einen Beitrag für den Umweltschutz.

3.3. Unabhängig davon, ob Sie einen Neukauf planen oder nicht: Was brauchen Sie Ihrer Meinung nach, um sich urteilssicher oder kompetent beim Einkauf und der Nutzung von vernetzten Geräten zu fühlen?

[5-stufige Skala mit: Stimme gar nicht zu; Stimme eher nicht zu; unbestimmt; Stimme eher zu; Stimme voll und ganz zu]

- Mehr Wissen zur sachgemäßen Wartung und Pflege.
- Mehr Informationen zur Haltbarkeit und Zuverlässigkeit.
- Mehr Angebote zur Beratung bei Fehlern und Funktionseinschränkungen.
- Mehr Rechte, um im Zweifelsfall Entschädigungen zu erhalten.
- Mehr Beratung und Information durch die Hersteller.
- Mehr Beratung und Information durch den Fachhandel.
- Ein einfach verständliches und praktisches Design.
- Frei verfügbare Handbücher sowie Aufbau- und Funktionspläne.
- Informationen darüber, wie lange die Software bei meinem Gerät unterstützt wird.
- Informationen darüber, was mit dem Softwareupdate nach Supportende geschieht.
- Mehr Aufklärung über die Umweltbelastungen, die durch meine Nutzung der digitalen Geräte entstehen, die aber außerhalb meines Wirkungskreises liegen.
- Sonstige: offenes Textfeld, 500 Zeichen

3.3 Welchen der folgenden Aussagen stimmen Sie zu oder nicht zu?

[5-stufige Skala mit: Stimme gar nicht zu; Stimme eher nicht zu; unbestimmt; Stimme eher zu; Stimme voll und ganz zu]

- Die Digitalisierung verbessert die meisten meiner Lebensbereiche.
- Die Digitalisierung ist wichtig für den gesellschaftlichen Fortschritt.
- Ich wünsche mir mehr Informationen darüber, welche Daten meine Geräte speichern und was mit den Daten geschieht.
- Ich pflege einen kritischen und selbstbewussten Gebrauch von Informationstechnik.
- Ich befürchte, dass ich zu viele Daten über mich preisgebe, wenn ich digitale Geräte nutze.
- Der Schutz meiner persönlichen Daten ist mir nicht wichtig.
- Moderne, digitale Geräte sind aus meiner Sicht größtenteils sicher.
- Die zunehmende Digitalisierung des Alltags macht mir Angst.
- Ich fühle mich durch Politik und Verbraucherschutz ausreichend geschützt vor möglichen negativen Folgen der Digitalisierung.
- Ich befürchte, mich in Zukunft immer weniger mit alltäglichen Geräten auszukennen, weil sie immer komplexer werden.

- Ich leiste einen Beitrag zum Umwelt- und Klimaschutz, indem ich Geräte möglichst lange nutze.
- Ich informiere mich über die Möglichkeiten, meinen ökologischen Fußabdruck zu reduzieren.
- Ich habe Verständnis dafür, dass Produkte nicht lange halten, weil dadurch „alte“ Produkte schneller vom Markt genommen werden.
- Ich bin verärgert, dass ich gezwungen werden, ein Produkt zu erneuern, weil die Software nicht mehr funktioniert.

3.4 Was ist für Sie die wichtigste Eigenschaft eines vernetzten bzw. smarten Geräts?
[offenes Format, Textfeld mit 500 Zeichen]

3.5 Wer trägt Ihrer Ansicht nach die Verantwortung für die Software auf Ihren Geräten und warum?
[offenes Format, Textfeld mit 500 Zeichen]

3.6. Was sollte die Politik tun, damit smarte Geräte länger halten und nicht aufgrund von Software-Problemen den Geist aufgeben?
[offenes Format, Textfeld mit 500 Zeichen]

Teil 4: Selbsteinschätzung und Soziodemographie

Nun folgen nur noch wenige Fragen zu Ihrer Person.

4.1 Wie schätzen Sie Ihr Wissen in Bezug auf folgende Aspekte ein?

[5-stufige Skala: Sehr gering, Eher gering, mittelmäßig, eher hoch, eher hoch]

- Allgemeines Verständnis von Informationstechnik und zur Vernetzung von Geräten
- Technischer Aufbau und Funktionen von vernetzten Geräten
- Pflege und Wartung von vernetzten Geräten
- Handhabung von Software-Updates und Upgrades
- IT-Sicherheit und Datenschutz
- Reparatur von Geräten
- Umgang mit Software-Problemen
- Auswahl der passenden Software
- Meine Rechte als Verbraucher*in
- Meine Pflichten als Verbraucher*in

4.2 Wie alt sind Sie?

__ Jahre [Eingabe möglich von 14 bis 75 Jahren]

4.3 Geschlecht

- weiblich
- männlich
- divers

4.4 Welchen höchsten Schul- bzw. Hochschulabschluss haben Sie?

- aktuell noch Schüler*in in einer allgemeinbildenden Schule

- kein Haupt-/Volksschulabschluss
- Haupt-/Volksschulabschluss (oder frühere 8-klassige Schule) oder Polytechnische Oberschule mit Abschluss 8. oder 9. Klasse
- Mittlere Reife/Realschulabschluss/Polytechnische Oberschule 10. Klasse
- allgemeine oder fachgebundene Hochschulreife/Abitur (Gymnasium bzw. EOS) ohne abgeschlossenes Hochschul-/Universitätsstudium
- abgeschlossenes Universitäts-/Hochschul- bzw. Fachhochschulstudium (bzw. Fachschulabschluss mit Anerkennung als Fachhochschulabschluss)

4.5 Wie hoch ist das monatliche Nettoeinkommen Ihres Haushalts insgesamt?

Damit ist die Summe gemeint, die sich ergibt aus Lohn, Gehalt, Einkommen aus selbständiger Tätigkeit, Rente oder Pension, jeweils nach Abzug der Steuern und Sozialversicherungsbeiträge. Rechnen Sie bitte auch die Einkünfte aus öffentlichen Beihilfen, Einkommen aus Vermietung, Verpachtung, Wohngeld, Kindergeld und sonstige Einkünfte hinzu.

- unter 1.000 Euro
- 1.000 bis unter 2.000 Euro
- 2.000 bis unter 3.000 Euro
- 3.000 bis unter 4.000 Euro
- 4.000 bis unter 5.000 Euro
- 6.000 Euro und mehr

5.5 Wie viele Personen wohnen insgesamt einschließlich Ihnen in Ihrem Haushalt?

- Ich wohne allein [Filter: nächste Frage überspringen]
- 2 Personen
- 3 Personen
- 4 Personen
- 5 Personen
- 6 Personen
- 7 Personen und mehr

5.6 Leben Kinder unter 18 Jahren in Ihrem Haushalt? Wenn ja, viele?

[Filter: Nur Wenn Person nicht alleine wohnt, Frage 5 der Soziodemografie, Antwort „ich wohne alleine“]

- keine Kinder
- 1 Kind
- 2 Kinder
- 3 Kinder
- 4 Kinder
- 5 Kinder und mehr

5.7 Sind Sie in Ihrem Haushalt diejenige Person, die hauptsächlich für die Anschaffung von Haushaltsgeräten zuständig ist?

- ja
- nein
- geteilte Zuständigkeit