Technische Mindestanforderung Bereich: Softwareentwicklung und Softwareanpassungen (Customizing)

Verantwortlich:

Robert Gladewitz

Kevin Glück

DBFZ Leipzig

Stand: 22.03.2023

Nächste Überprüfung: 04.2024

Gültigkeit: 04.2024

Inhalt

1 Allgemeines

1.1 Ziel des Dokuments und allgemeine Vorgaben

Dieses Dokument legt die technischen Mindestanforderungen für individuell im Rahmen einer Softwareentwicklung oder im Zusammenhang mit einer sonstigen Beschaffung (z.B. Messgeräte) gekoppelte Software im Rahmen von Ausschreibungen, Projekten und Aufträgen fest, die durch das DBFZ direkt oder indirekt finanziert werden. Abweichungen von diesen technischen Mindestanforderungen sind vertraglich und in vorheriger Absprache mit der IT festzulegen, eine Abweichung von dem Schriftformerfordernis ist nicht möglich. Schriftliche oder mündliche Änderungen durch den Lieferanten der Software mit z.B. Projektträgern oder Organen der Gesellschaft außerhalb der Geschäftsführung sind nicht zulässig.

Nachweislich höherwertige Komponenten als die Dargestellten können verwendet werden und werden in der Regel im Rahmen der Angebotswertung positiv bewertet.

Alle technischen Mindestanforderungen des DBFZ sind unter https://www.dbfz.de/tma verfügbar.

1.2 Verwendete Abkürzungen und Begriffsdefinitionen

ADS	Active Directory Services
BSD	Berkeley Software Distribution
CSS	Cascading Style Sheets
GPL	GNU General Public License
JSON	JavaScript Object Notation
LPGL	GNU Lesser General Public License
MIT	Massachusetts Institute of Technology
PHP	Hypertext Preprocessor
Update	Aktualisierung der Software im Rahmen bestehender oder geringwertig neuer Funktionalitäten mit dem Ziel der Fehlerbehebung oder ergonomischen Verbesserung.
Upgrade	Aktualisierung der Software mit dem Ziel neuer Funktionalitäten, die Fehlerbehebung oder ergonomische Verbesserung ist untergeordnet.

1.3 Geltungsbereich

Dieses Dokument beschreibt alle Vorgaben für Softwareentwicklungen, Softwareanpassung (Customizing), die durch das DBFZ beauftragt werden oder durch Übernahme aus vertretungsweise für das DBFZ durchgeführten Vorgängen in den Besitz des DBFZ übergehen sollen.

Grundsätzlich entstehen bei einer Entwicklung von Softwarekomponenten folgende Dokument- und Inhaltstypen:

- Quelltext (Source Code)
- Maschinencode (Ausführbarer Code / Bibliotheken)
- Visualisierungsdaten

- Konfigurationsdaten
- Bilder und Texte
- Designs und Designvorlagen
- Templates

Die Vorgaben in diesem Dokument beziehen sich auf alle genannten Dokumenttypen. Anpassungen, die nicht den Vorgaben aus diesem Dokument entsprechen, müssen dokumentiert, begründet und bei Beschaffung dem Angebot beigefügt werden. Bei Übernahme bestehender Software sind Abweichungen von den Vorgaben zum Zeitpunkt der Übergabe an das DBFZ dokumentiert beizufügen.

Spezielle lizenzrechtliche Ausschlüsse oder Anpassungen müssen im Vorfeld einer möglichen Beauftragung mit der IT am DBFZ abgestimmt werden und sind nur mit schriftlicher Bestätigung der IT oder der Geschäftsführung möglich. Im Rahmen von formalisierten Vergabeverfahren müssen entsprechende Anfragen an die Vergabestelle gerichtet werden.

Die hier definierten Vorgaben gelten auch für Webseitenprojekte, Cloudanwendungen und SaaS.

2 Vorgaben

Alle für das DBFZ im Rahmen beauftragter Softwareentwicklungen oder Softwareanpassungen erstellten Dokument- und Inhaltstypen sind als freie Inhalte (Open Content) zu liefern. Urheberrechtlich sichert der Lieferant damit ein zeitlich wie räumlich unbeschränktes, nichtausschließliches, einfaches und mit dem Liefergegenstand zusammen (z.B. Softwareverkauf durch DBFZ) übertragbares Nutzungsrecht zu.

Hinweis: Für Softwareanpassungen, wie zum Beispiel die Einrichtung von Gebäudeautomatisierungs Komponenten oder Prozessleitsystemen, gelten die Richtlinien aus diesem Dokument für alle angepassten Komponenten uneingeschränkt. Konfigurationsanpassungen, Customizing, Definitionen und Zeichnungen sind gleichwertig zu den in diesem Dokument definierten Inhaltstypen für die Softwareentwicklung anzuwenden.

2.1 Quelltexte und Inhaltstypen

Für entwickelte Softwarekomponenten oder Softwareanpassungen sind alle einschlägigen definierten Dokumente und Inhaltstypen vollständig zur Verfügung zu stellen (Quellcode, Open Source).

Geschlossene Codelieferungen (Closed Source) sind nur zulässig, wenn die Weiterentwicklung, Veröffentlichung oder Verwendungsrechte nicht beeinträchtigt sind, zum Beispiel im Fall proprietärer Module oder Bibliotheken. In diesem Fall ist der Lieferant verpflichtet, durch Lieferung einer vollständigen Dokumentation die Verwendung des Moduls, Anpassung und Weiterentwicklung sicherzustellen. In der Dokumentation muss die Verarbeitung und Herleitung der Daten sowie die Ergebnisse lückenlos dargelegt werden.

2.1.1 Programmiersprache

Bei Softwareentwicklungen ist die verwendete Programmiersprache im Vorfeld mit den Verantwortlichen des DBFZ abzusprechen, bzw. bei formalisierten Vergabeverfahren anzugeben.

Eine Inline-Dokumentation ist für alle Quelltexte, die dieses Instrument erlaubt, anzuwenden.

Ein abgrenzbares Softwareprojekt ist einheitlich in einer Programmiersprache oder -technologie umzusetzen. Zulässige Ausnahmen bilden hierbei produktabhängige Vorgaben wie zum Beispiel die Verwendung von TypoScript und PHP in Typo3 Projekten.

2.1.2 Verwendung von Laufzeitumgebungen oder Interpretern

Bei Verwendung von vorgefertigten Laufzeitumgebungen, Modulen oder Softwarekomponenten gelten Vorgaben aus diesem Dokument nur für die neu erstellten oder angepassten Inhalte.

Dies ist beispielsweise bei Embedded oder Speziallösungen (z.B. Gebäudeautomatisierung, Prozessleittechnik von Versuchsanlagen oder Einzellösungen in der Automatisierung) der Fall, da die Herstellerkomponenten nicht mitentwickelt werden und somit besonderen Lizenzen unterliegen können.

Für alle entwickelten Anpassungen gelten die Vorgaben aus diesem Dokument uneingeschränkt. Es muss sichergestellt sein, dass das DBFZ als Auftraggeber eine Weiterentwicklung oder Anpassung der Software und Systeme auf Basis der vorliegenden Entwicklungsdateien (Quelldateien) ohne zusätzlichen Aufwand, zusätzliche Softwarekomponenten, Lizenzkosten oder eingekaufte Dienstleistungen fortführen kann. Dies bedeutet, dass notwendige Laufzeitumgebungen, Interpreter oder Entwicklungssoftware mit den notwendigen Softwarekomponenten und deren Lizenzen Bestandteil der Lieferung sein müssen.

Es ist bei der Verwendung von Laufzeitumgebungen grundsätzlich eine vollständige Dokumentation aller verwendeten Versionen, Module und Komponenten bereit zu stellen. Dies gilt auch für hinterlegte Funktionalitäten, die in der bezogenen Softwareentwicklung oder Softwareanpassung nicht verwendet werden. Die Rechte an der weiteren Verwendung der Laufzeitumgebungen oder Interpreter sind Bestandteil der Lieferung.

2.1.3 Verwendung von Frameworks und Templates

Vor einer Neuentwicklung ist vom Bedarfsträger grundsätzlich zu prüfen, ob eine gewünschte Funktionalität nicht bereits durch vorhandene Frameworks und Templates vorgehalten werden kann. Bei der Verwendung von Frameworks und Templates sind folgende zusätzliche Vorgaben zu beachten:

2.1.3.1 Wartung / Aktualität

Eine Wartung muss über den geplanten Betriebs- bzw. Bezugszeitraum gewährleistet sein, z.B. durch Updates oder Upgrades. Upgrades innerhalb des Bezugszeitraums sind mit dem Kaufpreis abgegolten.

2.1.3.2 Bevorzugte Quellen für Templates und Frameworks

Unter Linux Distributionen und Windows werden die Frameworks durch den Hersteller/Distributor zur Verfügung gestellt. Unter Windows entspräche dies zum Beispiel dem .NET Framework und unter Linux, Templates wie Smarty, OpenJDK oder Apache Velocity.

Grundsätzlich sind durch Distributionen verwaltete Templates zu bevorzugen, wenn hierbei keine Funktionseinschränkungen bestehen. Es muss hierbei darauf geachtet werden, dass die Kompatibilität für neuere Versionen, die zum Beispiel durch Distributionsupdates und –Upgrades zur Verfügung gestellt werden, gewahrt wird.

2.1.3.3 Graphische Elemente und Oberflächen

Für die Entwicklung barrierefreier graphischer Oberflächen und Webseiten sind die unterschiedlichen Auflösungen und Formate zu beachten. Die grafische Oberfläche muss sich flexibel auf die verschiedenen Eingabegeräte wie Smartphone, Tablet, Laptop oder Desktopcomputer einstellen und somit auf allen Geräten uneingeschränkt verwendbar sein. Als Grundlage wird ein flexibles CSS Templatesystem wie Bootstrap oder Foundation vorausgesetzt. Bei dem Einsatz von Javascript ist auf vorgefertigte Bibliotheken wie jQuery oder vergleichbare Technologien zurückzugreifen.

2.1.3.4 Technologien

Grundsätzlich verwenden aktuelle Frameworks von Herstellern zur Verfügung gestellte Technologien. Aktuell sind folgende Technologien zu bevorzugen:

- Microsoft .Net 64Bit
- OpenJDK (>20) 64Bit
- SOAP / Webservices
- HTML5 / XHTML / JSON
- CSS3 / jQuery oder vergleichbare JavaScript-Bibliotheken
- BAC Net

2.1.3.5 Nicht zulässige Technologien

Nicht verwendet werden dürfen Technologien, deren Abkündigung bereits durch den Hersteller bekannt gegeben wurde. Insbesondere sind hier folgende Technologien zu nennen:

- Microsoft COM / DCOM
- Microsoft Active X
- RPC (Binärübertragung)
- Oracle Java 1.8 oder älter

Die Verwendung von abgekündigten Technologien ist als Ausnahme möglich, wenn vorhandene Projekte weiterentwickelt werden oder wenn vorhandene Hardwarekomponenten hierdurch weiterverwendet werden können. Die Verwendung muss möglichst frühzeitig im Planungsstadium, spätestens aber vor Beginn des Beschaffungsprozesses mit der IT des DBFZ abgesprochen werden.

2.1.3.6 Architekturen (64Bit)

Die Entwicklung neuer Anwendungen oder die grundlegende Aktualisierung vorhandener Anwendungen ist ausschließlich auf Basis von 64Bit Technologien umzusetzen. Hierbei ist zwingend die Verwendung von 64Bit Bibliotheken der Frameworks oder Entwicklungsumgebungen vorgeschrieben. Zulässige Architekturen:

- AMD64 (X64)
- Power64
- Mips64
- Sparc64
- ARM (64)

2.1.3.7 Schutz bei Versionsupdates von abhängigen Bibliotheken

Für die Erstellung von Software ist darauf zu achten, dass der Quelltext immer voll kompatibel mit neueren Versionen von verwendeten Bibliotheken ist. Für .Net, Java und PHP werden beispielsweise frühzeitig Funktionen als „Obsolete“ oder „Deprecated“ gekennzeichnet. Die Verwendung von Funktionen, die bereits während der Entwicklungsphase als „Obsolete“ oder „Deprecated“ gekennzeichnet werden, ist ausdrücklich nicht zulässig.

2.1.4 Programmierstil / Code Style

Für die Erstellung von Code ist der von der Programmiersprache empfohlene Programmierstil anzuwenden. Zusätzlich sind die Vorgaben aus der ISO/IEC9126 zu beachten und umzusetzen.

Wenn seitens des DBFZ, für zum Beispiel ein Projekt, ein eigener Programmierstil definiert wurde, ist dieser bei Auftragserteilung mitzuteilen und vom Auftragnehmer anzuwenden.

Temporäre Quellcodezeilen sind grundsätzlich mit deren Risiken im Quelltext direkt zu dokumentieren. Dies gilt auch bei Anwendung von Workarounds. Zusätzlich muss der Verantwortliche im DBFZ umgehend über die Anwendung solcher Veränderungen im Quelltext informiert werden.

2.1.5 Dokumentation

Zusätzlich zur Inline Dokumentation muss eine vollständige Dokumentation erstellt und dem DBFZ überlassen werden. Hierbei sind sowohl gedruckte als auch elektronische Dokumente zulässig. Die Aktualität der Dokumentation muss mit der Aktualität der übergebenen Software übereinstimmen.

Der Verweis auf eine Onlinedokumentation ist nur zulässig, wenn es sich um Standardprodukte wie Programmierumgebungen, Templates oder Frameworks handelt. Für die im Auftrag erzeugten Dokumenteninhalte ist dies ausdrücklich nicht zulässig.

Die folgende Tabelle beschreibt die Dokumentationsinhalte.

Dokumentationsteil	Zwingend erforderlich	Hinweise/Einschränkungen
Zielstellung	Ja	
Entwicklungsdokumentation (Funktionsdiagramme, etc.)	Ja	Zulässig sind alle gängigen Diagrammformen (Struktogramm, Jackson-Diagramm, Funktionsdiagramm, etc.)
Einsatzzweck	Optional	
Aufbau der Dateistruktur	Ja	
Dokumentation der zusätzlichen Module	Ja	Bei geschlossenem Code müssen alle Funktionen, auch die nicht in der Entwicklung verwendeten Module, dokumentiert werden
Methodendokumentation	Ja	Zusätzlich zur Inline Dokumentation
Installationsdokumentation	Ja	
Datendokumentation	Ja	Datentypen
Benutzerdokumentation	Ja	Dokumentation zur Verwendung, Für Endbenutzer.
Testdokumentation	Optional	

Tabelle 1: Inhalte für Dokumentationen

Grundsätzlich müssen sowohl Funktionen (Prozeduren, Subprogramme) als auch verwendete Variablen dokumentiert werden.

2.2 Verwendungsrechte

Für alle Softwareprojekte sichert der Ersteller/Auftragnehmer zu, dass nach Abgeltung der finanziellen Interessen das uneingeschränkte Verwendungsrecht für alle im Geltungsbereich definierten Dokument- und Inhaltstypen besteht.

Hierbei geht das DBFZ grundsätzlich davon aus, dass eine interne Weiterentwicklung oder eine Weiterentwicklung durch Dritte uneingeschränkt erlaubt ist.

Für Zusatzmodule mit Einschränkungen muss der Auftragnehmer das zeitlich uneingeschränkte Verwendungsrecht zusichern. Zeitliche Einschränkungen oder Einschränkungen auf Personen oder Geräte sind nicht zulässig.

2.3 Weitergabe

Das DBFZ behält sich die Weitergabe von den im Geltungsbereich definierten Dokument- und Inhaltstypen ohne Einschränkungen oder finanzieller Interessen des DBFZ vor. Die Bedingungen über Weitergabe obliegen dem DBFZ.

2.4 Veröffentlichung

Für alle Softwareprojekte gilt das uneingeschränkte Recht, die im Geltungsbereich definierten Dokument- und Inhaltstypen zu veröffentlichen. Das Recht beschränkt sich nicht auf wissenschaftliche Arbeiten.

2.5 Abweichend zulässige Lizenzformen für erstellten Code

Der Auftraggeber darf eingeschränkt die Lizenzform vorgeben. Die Lizenzierung ist mit folgenden Lizenzmodellen möglich:

Lizenz	Anmerkungen
GPL V2	Nur, wenn die Version 3 aufgrund verwendeter Komponenten nicht möglich ist
GPL V3	
BSD-Lizenz	
Apache Lizenz	Nur nach Absprache mit der Geschäftsführung, genaue vertragliche Bedingungen müssen definiert werden
LGPL	Nur nach Absprache mit dem Projektverantwortlichen
MIT-Lizenz	

Tabelle 2: zulässige Lizenzformen

Gesonderte Lizenzmodelle für verwendete Bibliotheken und notwendige Programme (zum Beispiel Entwicklungsumgebungen) sind zulässig, dürfen aber die Vorgaben dieses Dokuments für den neu erstellten Inhalt nicht beeinträchtigen.

2.6 Sicherheit

Bei der Softwareentwicklung von sicherheitskritischen Anwendungen und Diensten muss eine spätere Betriebs- und Wartungsphase zugesichert werden. Die Kosten müssen hierbei durch einen während der Ausschreibung erstellten Wartungsvertrag festgelegt werden.

2.6.1 Verschlüsselung

Für die Entwicklung von Software muss bei Kommunikationskomponenten eine als sicher geltende Verschlüsselung verwendet werden. Wenn eine vollständige Verschlüsselung

aufgrund der verwendeten Komponenten oder Techniken nicht möglich ist, muss zumindest die Authentifizierung mittels einer sicheren Verschlüsselung abgesichert werden. Eine Übertragung von Hash Authentifizierungsdaten (SHA etc.) ist nicht zulässig. Als sicher gelten aktuell folgende symmetrische Verschlüsselungsverfahren:

Verschlüsselung	Anmerkungen
AES (Rijndael)	Minimale Schüssellänge 128, empfohlen 256
Twofish	
MARS	
Serpent	

Tabelle 3: Symmetrische Verschlüsselungsarten

Grundsätzlich muss, wenn nicht technisch oder vertraglich anders vorgegeben, eine minimale Schlüssellänge von 256 Bit verwendet werden. Nicht zulässig sind die noch häufig verwendeten Verfahren RC4, DES und 3DES.

Für Anwendungen mit asymmetrischer Verschlüsselung sind folgende Verfahren zulässig:

- RSA (Schlüssellänge min. 2048 Bit)
- Elgamal

Eine Mischung von asymmetrischer und symmetrischer Verschlüsselung ist zulässig und gewünscht, wenn diese die Sicherheit der Software erhöht.

Für eine Schlüsselvereinbarung bei gesicherten Verbindungen müssen sichere Verfahren verwendet werden. Für SSL Verbindungen wird dies durch das Schlüsselaustauschprotokoll Diffie-Hellman-Schlüsselaustausch sichergestellt. Fest vergebene gemeinsame Schlüssel sind grundsätzlich nur zulässig, wenn die Technologie dies erfordert.

2.6.2 Prüfsummenverfahren für Kennwörter und Hashsummen

Für Prüfsummen werden folgende Vorgaben definiert.

Verfahren	Kennwörter	Dateien/Daten	Anmerkung
MD5	Nein	Ja	
**SHA1	Nein	Ja	
SHA2 (SHA256)	Ja	Ja	

Tabelle 4: Hashsummenarten

**SHA1 kann mit Ausnahme und nach Rücksprache mit der IT für Kennwörter verwendet werden, wenn dies durch Kompatibilitätseinschränkungen oder Einschränkungen in der Entwicklungsumgebung notwendig ist.

Zur Erhöhung der Sicherheit sollten nach Möglichkeit „Salt and Pepper“ Verfahren verwendet werden. Die Verwendung beider Verfahren zusammen ist ebenfalls zulässig.

2.7 Wartung und Mangelbeseitigung

Ein Angebot über einen Wartungsvertrag für die Softwarewartung ist mit dem Vertrag der Softwareentwicklung während eines Vergabeverfahrens dem Angebot beizufügen.

Grundsätzlich soll für alle Softwarekomponenten, für die eine Laufzeit zu erwarten ist, ein Wartungsvertrag abgeschlossen werden. Die Entscheidung über den Abschluss eines Wartungsvertrages obliegt dem DBFZ.

Für Webseiten oder technische Anlagen ist hier der Rahmenvertrag für die Wartung so zu definieren, dass Reaktions-, Service und Wiederherstellungszeiten festgelegt werden. Die Fehlerbeseitigung selbst muss ohne weitere Kosten mit einem Wartungsvertrag abgegolten sein.

In der folgenden Tabelle werden die vorzugebenen Reaktions- und Wiederherstellungszeiten definiert.

Mängelklasse	Reaktionszeit	Wiederherstellungszeit
Betriebsverhindernder Mangel	24 Stunden	48 Stunden
Betriebsbehindernder Mangel	24 Stunden	48 Stunden
Leichter Mangel	48 Stunden	eine Woche

Tabelle 5: Reaktions- und Wiederherstellungszeiten

2.8 Sicherheitslücken

Bei der Entwicklung oder Anpassungen von Software muss drauf geachtet werden, dass die aktuellste Version mit dem Status „stable“ oder eine LTS Version der darauf aufbauenden Templates oder Frameworks benutzt wird.

Sollten aktiv ausgenutzte Sicherheitslücken durch die Hersteller oder andere Institute wie das DFN-CERT bekannt werden, müssen diese umgehend geschlossen werden, spätestens aber nach 2 Tagen.

2.9 Verwendung von vertrauenswürdigen Repositories und Entwicklungsquellen

Grundsätzlich muss Entwicklung von Software unter Verwendung von distributionsgestützten Repositories umgesetzt werden. Hierbei können herstellergepflegte Repositories verwendet werden, wenn diese den Prinzipien der stabilen Softwarebereitstellung entsprechen.

Voraussetzung für solch eine Quelle ist, dass alle Bereitstellungsmechanismen veröffentlicht, geprüft, akkreditiert und standardisiert sind. Auch müssen nachweislich Security Report und Security Response Mechanismen mittels öffentlicher Bug-Reports und entsprechende CVE Veröffentlichungen bereitstellt werden.

Geschlossene Reporting- und Monitoringsysteme für Sicherheitsfehler sind ausdrücklich nicht zulässig.

Beispiele für vertrauenswürdige Repositories

Name	Gepflegt vom	Hinweis
Ubuntu	Distribution	Debianbasiert , Canonical, für LTS Lizenz notwendig
Debian	Distribution	
Nodesource	Hersteller/Community	`deb.nodesource.com`
RedHat	Distribution	
IBM	Hersteller	
Zabbix	Hersteller	

Tabelle 6: vertrauenswürdige Repositories

Die Freigabe von neuen Repositories muss durch die IT und den IT-Sicherheitsbeauftragten bestätigt werden. Eine Freigabe ist schriftlich im Ticketsystem des DBFZ zu dokumentieren und darf nur auf ein definiertes System erfolgen.

2.10 Einschränkungen für nicht vertrauenswürdige Quellen

Besondere Einschränkungen und Vorgaben gelten für die Verwendung von nicht vertrauenswürdigen Quellen. Als nicht vertrauenswürdige Quellen sind alle Quellen zu werten, bei denen kein dokumentiertes Monitoring und Sicherheitsreporting durchgeführt wird, nur teilweise CVEs veröffentlicht werden und auch keine DFN Cert Meldungen für Fehler und Bugs erstellt werden. Insbesondere gehören hierzu:

- GIT (alle Git Quellen, GitHub/GitLab)
- Docker (mit Ausnahme speziell signierter, vom Hersteller bereitgestellter Container)
- PIP
- PPA
- NPM (Javascript)
- PEAR (PHP)
- Anaconda, Miniconda
- Jupyter
- Private Quellen

Die genannten Quellen dürfen nur für interne Projekte verwendet werden, wenn eine Bereitstellung über Standardpakete der vertrauenswürdigen Repositories nicht möglich ist.

2.10.1 Nicht vertrauenswürdige Quellen für öffentliche Dienste (Public-Services)

Für Software, die bei veröffentlichten Anwendungen und Diensten verwendet wird (public services), ist eine Verwendung dieser Quellen unzulässig.

2.10.2 Nicht vertrauenswürdige Quellen für interne Dienste (Internes-Services)

Wenn nicht vertrauenswürdige Quellen für interne Projekte und Softwarevorhaben verwendet werden müssen, müssen die jeweilig verwendeten Stände (jede verwendete Version) in von der DBFZ bereitgestellten Speicherbereichen archiviert werden. Eine Verwendung ohne vorherige Archivierung muss ausgeschlossen werden.

Zudem müssen periodisch CVE Scans eingerichtet und durchgeführt werden. Sofern Softwareteile durch bekannte Schwachstellen kompromittierbar sind, müssen diese

umgehend durch Aktualisierung bereitgestellt und notwendigen Anpassungen umgesetzt werden.

Wenn dies nicht umgehend möglich ist, muss das DBFZ informiert werden und der betreffende Dienst deaktiviert oder die betroffene Software für die Nutzer gelöscht oder deaktiviert werden.