

Pflichtenheft 0.8 *Google Muddle*

Manuel Mästinger

Modularbeit 4.2 Software Engineering

ZHAW, School of Engineering, MAS Informatik 8

17. Juni 2015

Änderungsverlauf

Tabelle 1 zeigt Änderungen, die im Verlauf der Bearbeitung an diesem Dokument vorgenommen wurden.

Version	Beschreibung	Datum
0.1	Einleitung geschrieben	25.03.2015
0.2	Literatur und Abkürzungsverzeichnis eingefügt	25.03.2015
0.3	Grobstrukturierung des Dokuments	27.03.2015
0.4	Mit nichtfunktionalen Anforderungen begonnen	17.04.2015
0.5	Nichtfunktionale Anforderungen erweitert	10.05.2015
0.6	Funktionalen Anforderungen	16.05.2015
0.7	Überarbeitung des gesamten Dokuments	17.05.2015
0.8	Wenige Korrekturen und Ergänzungen	14.06.2015

Tabelle 1: Änderungen am Dokument

Inhaltsverzeichnis

1	Einleitung	4
1.1	Zum Dokument	4
1.2	Referenzdokumente	4
1.3	Motivation	4
1.4	Soll-Zustand	5
2	Funktionale Anforderungen	6
2.1	Browserstart	7
2.1.1	Auslöser	7
2.1.2	AkteurInnen	7
2.1.3	Essenzielle Schritte	7
2.2	Eruieren der Ausführung	8
2.2.1	Auslöser	8
2.2.2	AkteurInnen	8
2.2.3	Essenzielle Schritte	8
2.3	Suchanfrage	9
2.3.1	Auslöser	9
2.3.2	AkteurInnen	9
2.3.3	Essenzielle Schritte	9
2.4	Datenrücksendung	10
2.5	Update	10
2.6	Browserende	10
3	Nichtfunktionale Anforderungen	11
3.1	Interoperabilität	12
3.1.1	Beschreibung	12
3.1.2	Messverfahren	12
3.1.3	Geltungsbereich	12
3.2	Bedienbarkeit	13
3.2.1	Beschreibung	13
3.2.2	Messverfahren	13
3.2.3	Geltungsbereich	13
3.3	Verbrauchsverhalten	14
3.3.1	Beschreibung	14
3.3.2	Messverfahren	14
3.3.3	Geltungsbereich	14
3.4	Installierbarkeit	15
3.4.1	Beschreibung	15
3.4.2	Messverfahren	15
3.4.3	Geltungsbereich	15
4	Rahmenbedingungen	16

1 Einleitung

1.1 Zum Dokument

Dieses Dokument soll als möglichst präzise Vorlage für die Grobspezifikation der Applikation *Google Muddle* dienen.

Neben den funktionalen (Kapitel 2) und den nichtfunktionalen Anforderungen (Kapitel 3) werden in diesem Dokument auch die Rahmenbedingungen (Kapitel 4) festgelegt. Ergänzend sind im Kapitel 1 die Motivation sowie der angestrebte Soll-Zustand kurz skizziert.

Am Ende des Dokuments können dann die Verweise auf zitierte Literatur, das Abkürzungs- und das Tabellenverzeichnis gefunden werden.

Um die Geschlechtsneutralität der Aussagen zu gewährleisten werden in diesem Dokument in der Regel Binnenmajuskeln verwendet. So sollen die weibliche und die männliche Bezeichnung in kurzer Form vereint werden (vgl. <https://de.wikipedia.org/wiki/Binnen-I>).

1.2 Referenzdokumente

Es sind keine Referenzdokumente vorhanden.

1.3 Motivation

Wie die Enthüllungen durch Snowden (2013) gezeigt haben, werden (durch die NSA) flächendeckend Überwachungstechniken eingesetzt, die das Verhalten fast aller Internetbesucher protokollieren und automatisch auswerten. Multinationale Internetkonzerne wie Google mit strategisch gut geeignetem Dienstleistungsangebot und ausreichendem Kundenstamm spielen dabei eine zentrale Rolle. Um dieser globalen Überwachung zu entgehen, oder sie wenigstens zu erschweren, gibt es meines Erachtens drei Strategien, deren Kombination höchstmöglichen Schutz der Privatsphäre bietet.

Erstens kann versucht werden, Nutz- und Metadaten so zu verschlüsseln bzw. zu verschleiern (zum Beispiel durch die Umleitung über viele kaskadierte Knoten), dass deren Protokollierung nutzlos würde, da die Auswertung kaum mehr möglich wäre. Da diese Strategie sehr aufwendig ist und dazu noch von einer grossen Anzahl BenutzerInnen umgesetzt werden müsste, damit die jeweilige Kommunikation geschützt wäre, ist ihre baldige Verbreitung trotz vorhandener Werkzeuge meiner Einschätzung nach eher unwahrscheinlich.

Zweitens könnte der Versuch unternommen werden, die überwachenden Unternehmen und die überwachten Internetknoten komplett zu meiden. Dies würde faktisch zum totalen Verzicht auf das Internet führen, da praktisch der gesamte Datenverkehr über wenige einzelne (überwachte) Knoten verläuft. Zudem werden die am stärksten genutzten Dienstleistung von einigen we-

nigen Konzernen angeboten, die gemäss Snowden (2013) oft ebenfalls zur (geheimen) Überwachung verpflichtet sind.

Die dritte Strategie, auf welche ich in diesem Projekt zurückgreifen werde, funktioniert komplett anders. Anstatt die Nutz- und Metadaten so zu manipulieren, dass sie nicht mehr ausgewertet werden können oder auf grosse Teile des Internets zu verzichten, wird mit dieser Strategie versucht, die echten Daten durch grosse Mengen von künstlich erzeugten Daten zu erweitern, die sich kaum von den Echten unterscheiden lassen. Durch die entstehende Datenflut wird die Auswertung der Protokolle stark verfälscht und nutzlos.

1.4 Soll-Zustand

Um diese dritte Strategie umzusetzen, sind verschiedene Werkzeuge, die jeweils auf das verwendete Kommunikationsmittel bzw. die Dienstleistung angepasst sind, notwendig. Um einen grossen Bereich der Internetnutzung abzudecken, soll in diesem Projekt eine Erweiterung für Internet-Browser geschrieben werden, welche die Google Suche wie folgt nutzt.

In der gesamten Zeit, in der ein Browser mit dieser Erweiterung läuft, sollen im Hintergrund (ohne, dass der/die BenutzerIn dadurch gestört wird) Suchanfragen an Google gestellt werden. Die Suchanfragen sowie die zeitlichen Abstände zwischen einzelnen Anfragen sollen dabei möglichst nicht von echten Suchanfragen unterschieden werden können. Anschliessend soll ein zufällig ausgewähltes Suchresultat geöffnet werden. Als Quelle für Suchbegriffe könnten zum Beispiel echte Suchanfragen anderer BenutzerInnen der Browser-Erweiterung dienen.

Diese Methode hat ausserdem den positiven Nebeneffekt, die so genannte Filterblase zu verhindern. Unter Filterblase wird der Mechanismus verstanden, dass Suchmaschinen BenutzerInnenprofile anlegen und den jeweiligen BenutzerInnen auf sie zugeschnittene Ergebnisse anzeigen. Dabei entgehen den Leuten wichtige Alternativen, die vom Algorithmus bereits ausgefiltert wurden.

Dabei soll sich diese Arbeit ausschliesslich mit dem Client-Teil der Software beschäftigen. Allfällige Server-Komponenten sollen vorerst nicht berücksichtigt werden.

2 Funktionale Anforderungen

Folgende Tabelle (2) zeigt eine Übersicht aller funktionalen Anforderungen an *Google Muddle*.

Die funktionalen Anforderungen legen in ihrer Gesamtheit fest, welche Aufgaben die Applikation zu erfüllen hat. Dazu werden neben den Auslösern auch mögliche AkteurInnen und die essenziellen Schritte aufgeführt, aus welchen sich die jeweilige Anforderung zusammensetzt. Die funktionalen Anforderungen gehen dabei nicht auf die Eigenschaften ein, welche die Applikation neben der Funktionalität auch noch zu erfüllen hat. Sie werden in Kapitel 3 definiert.

In der Grobspezifikation sollen aus diesen Anforderungen die Anwendungsfälle abgeleitet werden können.

Nummer	Bezeichnung	Quelle
2.1	Browserstart	Autor des Dokuments
2.2	Eruieren der Ausführung	Autor des Dokuments
2.3	Suchanfrage	Autor des Dokuments
2.4	Datenrücksendung	Autor des Dokuments
2.5	Update	Autor des Dokuments
2.6	Browserende	Autor des Dokuments

Tabelle 2: Übersicht der funktionalen Anforderungen

2.1 Browserstart

Da die Applikation nicht direkt mit dem/der BenutzerIn interagiert, soll er/sie auch nicht durch sie gestört werden (siehe dazu die nichtfunktionale Anforderung 3.3). Es ist daher essenziell, dass gerade beim Start der Applikation wenig Ressourcen (Rechenleistung und Netzwerkverbindung) verbraucht werden. Dies bedingt, dass zu diesem Zeitpunkt nur wenig gemacht wird.

Es wird also lediglich die Funktion¹ gestartet, welche dauerhaft im Hintergrund läuft, alles andere wird bei entsprechender Ressourcenverfügbarkeit später (in ebendieser Funktion) erledigt.

2.1.1 Auslöser

Ausgelöst wird diese Funktion indirekt durch den/die BenutzerIn, welcher den Browser (mit *Google Muddle* als Erweiterung) startet. Direkter Auslöser ist somit der Browser.

2.1.2 AkteurInnen

Der/die BenutzerIn ist indirekt AkteurIn, da er/sie den Browser startet, welcher wiederum *Google Muddle* startet.

2.1.3 Essenzielle Schritte

1. Der/die BenutzerIn startet den Browser.
2. Der Browser initialisiert seine Erweiterungen, dabei wird auch *Google Muddle* gestartet.
3. Die “Eruierung der Ausführung” (funktionale Anforderung 2.2) wird gestartet.

¹Funktion meint in diesem Zusammenhang lediglich die als funktionale Anforderung gruppierten Ausführungsschritte und hat nichts mit dem Konstrukt der Funktion einer Programmiersprache zu tun. Ob eine solche funktionale Anforderung später als Funktion implementiert wird ist Sache der Software-Architektur.

2.2 Eruiere der Ausführung

Mit “Eruiere der Ausführung” ist gemeint, dass die Applikation ständig überprüft, wie stark der Computer und die Netzwerkverbindung gerade ausgelastet sind und, wenn genügend Ressourcen zur Verfügung stehen, Suchanfragen oder Updates initiiert.

2.2.1 Auslöser

Auslöser ist der Start der Applikation (funktionale Anforderung 2.1).

2.2.2 AkteurInnen

AkteurInnen sind für diese funktionale Anforderung keine vorhanden.

2.2.3 Essenzielle Schritte

1. Die Applikation wartet eine zufällige Zeit zwischen 5 Sekunden und 2 Minuten¹.
2. *Google Muddle* überprüft, ob der Computer und die Netzwerkverbindung ausgelastet sind. Falls eine Schwelle überschritten wird, die für den/die BenutzerIn bemerkbar ist, wird Punkt 1 erneut ausgeführt. Um hier eine Endlosschleife zu vermeiden, soll die Schwelle dynamisch an die Gegebenheiten angepasst werden.
3. Falls die letzte Überprüfung schon über 7 Tage² her ist überprüft *Google Muddle*, ob eine neuere Version verfügbar ist und installiert diese gegebenenfalls (funktionale Anforderung 2.5). Dabei wird der Zeitpunkt dieser Überprüfung protokolliert. Sollte eine neuere Version installiert worden sein, wird *Google Muddle* neu gestartet.
4. Eine Suchanfrage wird gestartet (funktionale Anforderung 2.3).
5. Die Schritte werden erneut ab Punkt 1 ausgeführt.

¹Bei diesen Zeiten handelt es sich um Vorschläge, die im Verlauf der Entwicklung durch Erfahrungswerte ersetzt werden sollten.

²Die 7 Tage werden als angemessen eingeschätzt. Auch hier darf die Zahl im Verlauf der Entwicklung gerne angepasst werden, wenn die Erfahrung zeigt, dass ein anderer Wert optimal wäre.

2.3 Suchanfrage

Soll eine neue Suchanfrage gestellt werden, wird zuerst auf die Datenbank mit bisherigen Suchbegriffen aller Benutzer von *Google Muddle* zugegriffen. Dabei stellt der Server einen Suchbegriff zur Verfügung (siehe 2.3.3 Essenzielle Schritte). Nachdem nach diesem Begriff gesucht wurde, soll ein zufällig ausgewähltes Ergebnis geöffnet werden. Dies lässt die Suche “echter” wirken und verfälscht weitere Statistiken.

2.3.1 Auslöser

Auslöser ist das “Eruieren der Ausführung” (funktionale Anforderung 2.2) beziehungsweise die Zeit, da immer wieder (in unterschiedlichen zufälligen Zeitintervallen) Suchanfragen ausgelöst werden.

2.3.2 AkteurInnen

AkteurInnen sind für diese funktionale Anforderung keine vorhanden.

2.3.3 Essenzielle Schritte

1. Herunterladen eines Suchbegriffes. Der Suchbegriff wird vom Server auf zufälliger Basis zur Verfügung gestellt, ohne, dass der Client (*Google Muddle*) darauf Einfluss nehmen könnte.¹
2. Bei Google nach dem erhaltenen Begriff (beziehungsweise dem Satz) suchen.
3. Ein von Google zur Verfügung gestelltes Suchresultat nach dem Zufallsprinzip² öffnen.
4. Die Verbindung schliessen.

¹Dies ermöglicht es, später serverseitig neue Features einzubauen. So könnte zum Beispiel der reine Zufall durch einen besseren Algorithmus ersetzt werden, der reale Trends abzuflachen versucht oder vermehrt nach indizierten Wörtern sucht, um diesen Überwachungsmechanismus zu stören.

²Auch dieser Zufallsmechanismus könnte später durch einen “intelligenten” Algorithmus ersetzt werden.

2.4 Datenrücksendung

Mit Datenrücksendung ist gemeint, dass echte Suchanfragen der BenutzerInnen an eine Datenbank übermittelt werden, woraus sie dann für automatische Suchanfragen verwendet werden können.

Diese funktionale Anforderung wird im Rahmen der Modularbeit nicht erfasst.

2.5 Update

Der Update wird immer dann ausgeführt, wenn eine neuere Version von *Google Muddle* zur Verfügung steht.

Diese funktionale Anforderung wird im Rahmen der Modularbeit nicht erfasst.

2.6 Browserende

Diese Funktion tritt in Kraft, wenn der Browser geschlossen wird.

Sie wird im Rahmen der Modularbeit nicht erfasst.

3 Nichtfunktionale Anforderungen

Eine Anforderung ist gemäss Glinz (2006) als nichtfunktional zu verstehen, wenn “das ihr zu Grunde liegende Bedürfnis eine nicht gegenständliche Eigenschaft [der Applikation] ist”. In diesem Kapitel werden also Anforderungen definiert, bei welchen das (der Anforderung zu Grunde liegende) Bedürfnis nicht die Funktionalität selbst ist.

In Tabelle 3 ist die (vollständige) Liste der Qualitätsmerkmale von Software gemäss ISO/IEC 9126 (DIN 66272) zu finden. Einige davon werden im Folgenden als nichtfunktionale Anforderungen in Bezug auf *Google Muddle* genauer beschrieben. Die übrigen werden im Rahmen dieser Modularbeit nicht erfasst.

Kategorie	Qualitätsmerkmal	Priorität
Funktionalität:	Angemessenheit	mittel
	Richtigkeit	hoch
	Interoperabilität (3.1)	hoch
	Sicherheit	hoch
	Ordnungsmässigkeit	mittel
	Konformität	gering
Zuverlässigkeit:	Reife	mittel
	Fehlertoleranz	hoch
	Wiederherstellbarkeit	gering
	Konformität	gering
Benutzbarkeit:	Verständlichkeit	mittel
	Erlernbarkeit	gering
	Bedienbarkeit (3.2)	hoch
	Attraktivität	hoch
	Konformität	gering
Effizienz:	Zeitverhalten	mittel
	Verbrauchsverhalten (3.3)	hoch
	Konformität	gering
Wartbarkeit/Änderbarkeit:	Analysierbarkeit	hoch
	Modifizierbarkeit	hoch
	Stabilität	hoch
	Testbarkeit	mittel
	Konformität	gering
Übertragbarkeit:	Anpassbarkeit	hoch
	Installierbarkeit (3.4)	hoch
	Koexistenz	gering
	Austauschbarkeit	gering
	Konformität	gering

Tabelle 3: Qualitätsmerkmale nach ISO/IEC 9126-1 (2001)

3.1 Interoperabilität

Interoperabilität bezeichnet gemäss ISO/IEC 9126-1 (2001) die “Fähigkeit, mit vorgegebenen Systemen zusammenzuwirken”.

3.1.1 Beschreibung

Es wird ein hoher Verbreitungsgrad der Applikation angestrebt, da sich dadurch ihre Wirkung verbessert. Deshalb ist es wichtig, eine möglichst breite Variation von Browsern zu unterstützen. Die Applikation soll mit den in der Tabelle 4 aufgelisteten Browsern (gemäss Gewichtung) in jeweils neuester Version (für Desktop-Betriebssysteme) kompatibel sein. Die Gewichtung wurde auf Grundlage der Browserstatistik von W3Schools (2015) vorgenommen.

Browser	Betriebssysteme	Gewichtung g
Apple Safari	Mac	1
Google Chrome	Linux, Mac, Windows	4
Internet Explorer	Windows	2
Mozilla Firefox	Linux, Mac, Windows	3
Opera	Mac, Windows	1

Tabelle 4: Liste kompatibler Browser mit Gewichtung

3.1.2 Messverfahren

Zwecks Messbarkeit wurden drei Stufen der Erfüllung, die pro Browser ermittelt werden, definiert (siehe Tabelle 5).

Erfüllungsgrad e	Beschreibung
1	Die Applikation läuft mit diesem Browser.
2	Die Applikation läuft mit diesem Browser und bietet einen reduzierten Funktionsumfang.
3	Die Applikation läuft mit diesem Browser und bietet den vollen Funktionsumfang.

Tabelle 5: Erfüllungsgrad der Interoperabilität pro Browser

Damit die Anforderung als erfüllt betrachtet wird, ist eine Summe für alle Browser aus Tabelle 4 aus Gewichtung g multipliziert mit Erfüllungsgrad e (also $\sum g * e$) von mindestens 29 erforderlich. Maximal ist 33 möglich.

3.1.3 Geltungsbereich

Diese Anforderung bezieht sich auf alle funktionalen Anforderungen, falls der Erfüllungsgrad 3 ist. Ansonsten (Erfüllungsgrad 1 und 2) kann die Untermenge beliebig ausgewählt werden (siehe Tabelle 5).

3.2 Bedienbarkeit

Gemeint ist gemäss ISO/IEC 9126-1 (2001) der “Aufwand für [...] BenutzerInnen], die Anwendung zu bedienen”.

3.2.1 Beschreibung

Da *Google Muddle* vollständig automatisch funktionieren muss, ist keine Interaktion mit BenutzerInnen gewünscht. Sie sollen auch nicht die Möglichkeit haben, Konfigurationen vorzunehmen, da die Verhaltensweise der Applikation für jedeN identisch sein soll. Dies vereinfacht die Weiterentwicklung vom Verhalten aufgrund neuer Erkenntnisse, da keine BenutzerInnenspezifische Einstellungen berücksichtigt werden müssen. So können auch jeweils die optimalen Einstellungen zur bestmöglichen Erreichung der Ziele (vgl. Kapitel 1.3) für alle gesetzt werden. Auch spezielle Funktionen wie Updates sollen automatisch - also ohne BenutzerInneninteraktion - stattfinden.

3.2.2 Messverfahren

So viele BenutzerInnen wie möglich werden die Applikation installiert. Nach der Installation versuchen sie, irgendwie mit der Applikation zu interagieren. Dies darf weder beim Start noch beim Beenden des Browsers möglich sein. Auch während der gesamten Laufzeit des Browsers oder bei einem allfälligen speziellen Funktion wie zum Beispiel einem Update darf die Erweiterung nicht wahrgenommen werden. (Als Abgrenzung zum Verbraucherverhalten (vgl. Kapitel 3.3), wo sie aufgrund des Ressourcenverbrauchs nicht wahrgenommen werden darf, gilt hier einzig die Wahrnehmung aufgrund von BenutzerInneninteraktionen wie zum Beispiel Konfiguration oder Fehlermeldungen.)

3.2.3 Geltungsbereich

Diese Anforderung gilt für die gesamten Laufzeit der Applikation.

3.3 Verbrauchsverhalten

Verbrauchsverhalten definiert nach ISO/IEC 9126-1 (2001) die “Anzahl und Dauer der benötigten Betriebsmittel bei der Erfüllung der Funktionen”.

3.3.1 Beschreibung

Da *Google Muddle* für seine BenutzerInnen direkt funktional wenig bietet, soll die Applikation auch möglichst unaufdringlich und ressourcenschonend arbeiten. Es ist nicht sehr relevant, wann die einzelnen Funktionen genau ausgeführt werden. Dies sollte daher möglichst dann geschehen, wenn der Computer wenig zu tun hat - also etwa wenn niemand damit am arbeiten ist.

3.3.2 Messverfahren

So viele BenutzerInnen wie möglich werden in einer Testphase mit einem Browser arbeiten, ohne zu wissen, ob die Applikation installiert ist oder nicht. Dabei dürfen sie keinen Unterschied feststellen. Die BenutzerInnen werden sowohl mit aktueller, als auch mit alter Hardware arbeiten. Auch die Betriebssysteme und die darauf installierten Browser werden divers sein (siehe Tabelle 4 auf Seite 12).

Wenn mehr als 5 Prozent der BenutzerInnen eindeutig erkennen, ob die Applikation installiert ist oder nicht, gilt diese Anforderung als nicht erfüllt.

3.3.3 Geltungsbereich

Diese Anforderung gilt für die gesamten Laufzeit der Applikation. Als besonders kritisch gilt dabei meist der Start, da in dieser Phase bewusst gewartet wird.

3.4 Installierbarkeit

ISO/IEC 9126-1 (2001) definiert die Installierbarkeit als “Aufwand, der zum Installieren der Software in einer festgelegten Umgebung notwendig ist”.

3.4.1 Beschreibung

Die Applikation soll im für den jeweiligen Browser offiziellen Store für Erweiterungen verfügbar sein.

Ist dies aufgrund von Einschränkungen oder Ausschluss seitens der BetreiberInnen dieses Stores nicht möglich, soll die Applikation über andere Plattformen zum herunterladen angeboten werden. Falls auf diese Möglichkeit zurückgegriffen werden muss, soll auf der Webseite dieser Plattform zusätzlich zur Applikation eine einfache Installationsanleitung präsentiert werden.

3.4.2 Messverfahren

Google Muddle muss - in jedem der Browser aus Tabelle 4 auf Seite 12 - über eine via Menü erreichbare Suchfunktion für Erweiterungen gefunden werden können. Anschliessend soll sie, über den für den jeweiligen Browser üblichen Weg, installiert werden können.

Ist die Applikation installiert, soll sie, ebenfalls über den für den jeweiligen Browser üblichen Weg, vollständig deinstalliert werden können.

Wird auf die oben erwähnte Ausweichmöglichkeit einer anderen Plattform zurückgegriffen, wird durch Rückfragen bei den BetreiberInnen geprüft, ob dies wirklich aufgrund von Einschränkungen oder Ausschluss seitens der BetreiberInnen geschehen ist. Sollte das nicht der Fall (gewesen) sein, gilt diese Anforderung als nicht erfüllt.

3.4.3 Geltungsbereich

Die Installierbarkeit hat keine Beziehung zu irgend einer funktionalen Anforderung.

4 Rahmenbedingungen

Die weiteren Bedingungen werden im Rahmen der Modularbeit nicht erfasst.

Literatur

Glinz, Martin. (2006). Requirements Engineering I - Nicht-funktionale Anforderungen. Heruntergeladen von https://files.ifi.uzh.ch/rerg/arvo/ftp/re_I/Kapitel_11_NFAnf.pdf am 27.03.2014

ISO/IEC 9126-1 (2001). Software engineering - Product quality - Part 1: Quality model. International Organization for Standardization.

Snowden, Edward. (2013). NSA interne Dokumente. Heruntergeladen von <https://edwardsnowden.com/2013/10/14/miscellaneous-leaked-documents/> am 25.03.2014

W3Schools (2015). Browser Statistics. Heruntergeladen von http://www.w3schools.com/browsers/browsers_stats.asp am 17.04.2014

Abkürzungen

DIN	Deutsches Institut für Normung
IEC	International Electrotechnical Commission, ein Normungsgremium für Elektrotechnik
ISO	International Organization for Standardization
NSA	National Security Agency der USA
USA	United States of America

Tabellenverzeichnis

1	Änderungen am Dokument	2
2	Übersicht der funktionalen Anforderungen	6
3	Qualitätsmerkmale nach ISO/IEC 9126-1 (2001)	11
4	Liste kompatibler Browser mit Gewichtung	12
5	Erfüllungsgrad der Interoperabilität pro Browser	12