Qualitätssicherung

ZWL-Roboter
HTW Berlin

Autor: Gruppe ZWL
Letzte Änderung: 12. Jan 2023
Dateiname: ZWL-Roboter_Qualitätssicherung(Version 0.9).docx
Version: 0.9

Copyright

Version Historie

Version	Datum	Verantwortlich	Änderung
0.1	10.12.2022	Markus	Programmcode Ausführen
0.2	11.12.2022	Markus	Farben erkennen und umwandeln.
0.3	14.12.2022	Markus	Servo-Motoren Ansteuerung
0.4	23.12.2022	Elian	Servo-Motoren mit 3D-Teile Ansteuerung
0.5	07.01.2023	Elian	Servo-Motor FS90R mit bestimmte Winkel Ansteuerung
0.6	10.01.2023	Elian	Servo-Motor SG90 mit bestimmte Winkel Ansteuerung
0.7	11.01.2023	Rayen	Kamera testen
0.8	11.01.2023	Rayen	Farbe erkennen und nach einem Kociemba string umwandeln
0.9	11.01.2023	Rayen	Ein Kociemba String Lösen
1.0			

Inhaltsverzeichnis

Verzeichnis vorhandener Dokumente II

1 Testfälle 3
1.1 Testfall 1: Programmcode Ausführen 3
1.2 Testfall 2: Farben erkennen und umwandeln. 4
1.3 Testfall 3: Servo-Motoren Ansteuerung 5
1.4 Testfall 4: Servo-Motoren mit 3D-Teile Ansteuerung 6
1.5 Testfall 5: Servo-Motor FS90R mit bestimmte Winkel Ansteuerung 7
1.6 Testfall 6: Servo-Motor SG90 mit bestimmte Winkel Ansteuerung 8
1.7 Testfall 7: Kamera testen 9
1.8 Testfall 8: Farbe erkennen und nach einem Kociemba string umwandeln 10
1.9 Testfall 9: Ein Kociemba String Lösen 11

2 Testprotokoll 13

Anhang 14
A Fehlerkategorien 14
B Qualitätskriterien nach ISO 9126 15
C Qualitätskriterien für Dokumente 16

... und in einen Kociemba-String umwandeln

Abbildungsverzeichnis

Abbildung 1: Programmcode Ausführe..... 3
Abbildung 2: Farben erkennen und umwandeln..... 4
Abbildung 3: Servo-motoren mit 3D-Teile Ansteuerung..... 6

Verzeichnis vorhandener Dokumente

Alle für die vorliegende Spezifikation ergänzenden Unterlagen müssen hier aufgeführt werden

Dokument	Autor	Datum
Automatisiertes Lösen des Zauberwürfels.docx	Gruppe ZWL	27.10.2022
ZWL-Roboter_Pflichtenheft.docx	Gruppe ZWL	24.11.2022
ZWL-Roboter_Technische_Spezifikation(Version 1.0).docx	Gruppe ZWL	15.12.2022
ZWL-Roboter_Qualitätssicherung(Version 1.0).docx	Gruppe ZWL	15.12.2022
Projektplan (Version 1.0).docx	Gruppe ZWL	24.11.2022

1 Testfälle

1.1 Testfall 1: Programmcode Ausführen

Testfall	Beschreibung
Testfall-Nummer	00001
Testart	Funktionstest
Opencv starten	Terminal Aufrufen, Visual-Studio Starten, Pakete laden, Code Ausführen
Testziel	Das Fenster der GUI für die Initialisierung soll gestartet werden, um weitere Eingaben zu tätigen.
Testvoraussetzungen	• Gültige Installation • Bibliotheken eingebunden • Hardware angeschlossen
Testfalldaten	-
Erwartetes Verhalten	Ein Fenster wird angezeigt

Testergebnis	□ Bestanden **X** Nicht Bestanden	
Fehlerkategorie	□ Leicht **X** Mittel □ Schwerwiegend	
Bemerkung	Code Format ist zwar kompilierbar aber die imports werden im System nicht gefunden, somit ist das Starten nicht möglich.	
Tester Kunde	Tester Auftragnehmer	Datum 10.12.2022

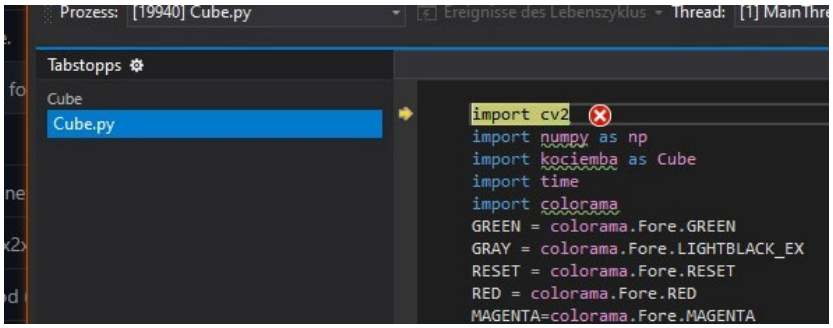

Abbildung 1: Programmcode Ausführe

1.2 Testfall 2: Farben erkennen und umwandeln.

Testfall	Beschreibung
Testfall-Nummer	00002
Testart	Funktionstest
opencv starten und Farben detektieren	Code ausführen, GUI anzeigen lassen und Farben suchen.
Testziel	In der GUI wird das Feld, wo der Würfel platziert, wird gezeigt und wenn man es dort platziert, sollen die Farben in eine Matrix gespeichert werden.
Testvoraussetzungen	• Gültige Installation • Bibliotheken eingebunden • Hardware angeschlossen (Camera) • Belichtung ist normal (nicht dunkel) • Objekt innerhalb des rasters
Testfalldaten	3x3 Matrix der Farben
Erwartetes Verhalten	Ein Cube wird mit den vorhandenen Farben ausgefüllt und die Werte werden in eine Matrix gespeichert

Testergebnis	**X** Bestanden □ Nicht Bestanden	
Fehlerkategorie	□ Leicht **X** Mittel □ Schwerwiegend	
Bemerkung	Wenn das Licht, der Abstand zum Objekt oder die Kamera schlecht / nicht optimal sind entstehen fehler in der Auslesung der Farben.	
Tester Kunde	Tester Auftragnehmer	Datum 11.12.2022

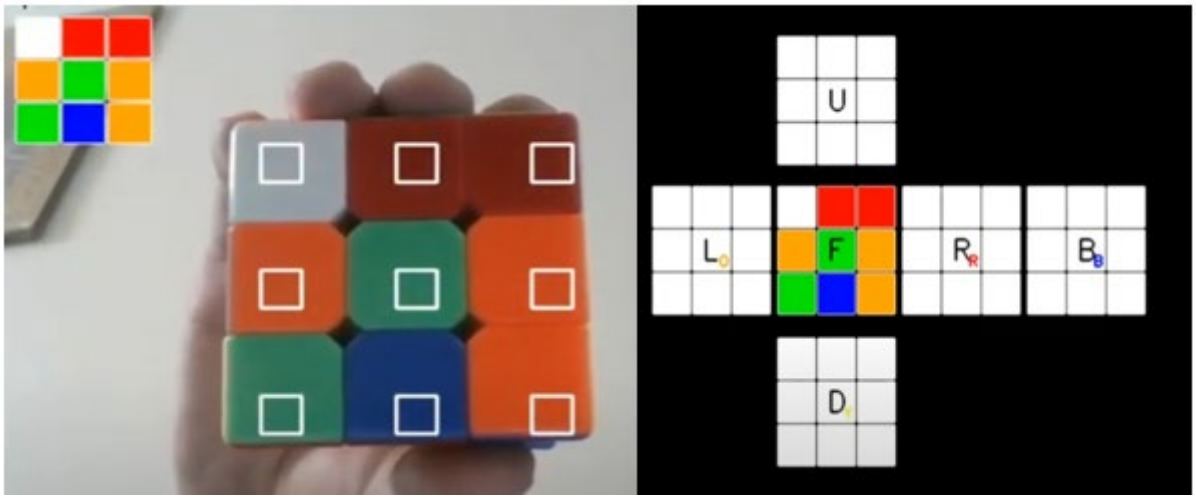

Abbildung 2: Farben erkennen und umwandeln

S

1.3 Testfall 3: Servo-Motoren Ansteuerung

Testfall	**Beschreibung**
Testfall-Nummer	00003
Testart	Funktionstest
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Arduino, Servo-Motoren, Arduino-IDE Software
Testziel	Den Servos anzusteuern, damit diese sich nicht gegenseitig behindern
Testvoraussetzungen	• Arduino IDE Compiler • Servo Motoren • Arduino Uno
Testfalldaten	Motoren drehen sich getrennt.
Erwartetes Verhalten	Drehung nacheinander und nie in der gleichen Zeit zusammen.

Testergebnis	**X** Bestanden □ Nicht Bestanden	
Fehlerkategorie	□ Leicht **X** Mittel □ Schwerwiegend	
Bemerkung	Sofern die Geschwindigkeit der Rotation nicht zu hoch ist, entstehen keine Fehler in der Bewegung.	
Tester Kunde	Tester Auftragnehmer	Datum 13.12.2022

1.4 Testfall 4: Servo-Motoren mit 3D-Teile Ansteuerung

Testfall	**Beschreibung**
Testfall-Nummer	00004
Testart	Funktionstest
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Arduino, Servo-Motoren,3D-Teile Arduino-IDE Software
Testziel	Den Servo-motor schafft den 3d-teile zu bewegen.
Testvoraussetzungen	• Arduino IDE Compiler • Servo Motoren • Arduino Uno R3 • 3D-Teil
Testfalldaten	-
Erwartetes Verhalten	-

Testergebnis	**X** Bestanden □ Nicht Bestanden	
Fehlerkategorie	□ Leicht **X** Mittel □ Schwerwiegend	
Bemerkung	Sofern die Geschwindigkeit der Rotation nicht zu hoch ist, entstehen keine Fehler in der Bewegung.	
Tester Kunde	Tester Auftragnehmer	Datum 23.12.2022

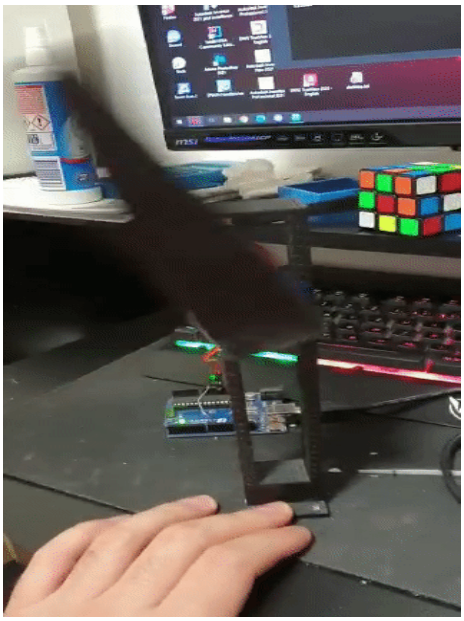

Abbildung 3: Servo-motoren mit 3D-Teile Ansteuerung

1.5 Testfall 5: Servo-Motor FS90R mit bestimmte Winkel Ansteuerung

Testfall	**Beschreibung**
Testfall-Nummer	00005
Testart	Funktionstest
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Arduino, Servo-Motoren(FS90R), Arduino-IDE Software
Testziel	Den Servo-motor mit bestimmte Winkel Ansteuerung (90° grad).
Testvoraussetzungen	• Arduino IDE Compiler • Servo Motoren • Arduino Uno R3
Testfalldaten	FS90R Motor dreht sich nicht um 90 Grad, sondern mehr als 90 grad
Erwartetes Verhalten	-

Testergebnis	□ Bestanden **X** Nicht Bestanden	
Fehlerkategorie	□ Leicht **X** Mittel □ Schwerwiegend	
Bemerkung	FS90R Motor dreht sich jedes Mal anderes, man versteht nicht, wie soll man mit den FS90R Motor umgehen	
Tester Kunde	Tester Auftragnehmer	Datum 07.01.2023

1.6 Testfall 6: Servo-Motor SG90 mit bestimmte Winkel Ansteuerung

Testfall	**Beschreibung**
Testfall-Nummer	00006
Testart	Funktionstest
Zu testender Geschäftsprozess/ Zu testende Funktionsgruppe	Arduino, Servo-Motoren (SG90), Arduino-IDE Software
Testziel	Den Servo-motor mit bestimmte Winkel Ansteuerung (90° grad).
Testvoraussetzungen	• Arduino IDE Compiler • Servo Motoren • Arduino Uno R3
Testfalldaten	FS90R Motor dreht sich um 90 Grad, kann nur von 0 bis 180 Grad Drehen
Erwartetes Verhalten	-

Testergebnis	**X** Bestanden □ Nicht Bestanden	
Fehlerkategorie	□ Leicht **X** Mittel □ Schwerwiegend	
Bemerkung		
Tester Kunde	Tester Auftragnehmer	Datum 10.01.2023

1.7 Testfall 7: Kamera testen

Testfall	**Beschreibung**	
Testfall-Nummer	00007	
Testart	Droidcam öffnen	
Befehlszeile ausführen	#bin/sh droidcam	
Testziel	Kamera testen	
Testvoraussetzungen	• Kamera • Droidcam im Handy installiert • Droidcam im Betriebssystem installiert	
Testfalldaten	IP Adresse von Handy, Droidcam port (4747)	
Erwartetes Verhalten	-> Telefon über Port 4747 mit dem Linux-Rechner verbinden, um auf Droidcam zuzugreifen -> Alles, was die Kamera aufzeichnet, muss auf dem Computerbildschirm sichtbar sein	
Testergebnis	**X** Bestanden ☐ Nicht Bestanden	
Fehlerkategorie	☐ Leicht **X** Mittel ☐ Schwerwiegend	
Bemerkung	Das Video vom Telefon kommt mit Verzögerung	
Tester Kunde	Tester Auftragnehmer	Datum 11.01.2022

1.8 Testfall 8: Farbe erkennen und nach einem Kociemba string umwandeln

Testfall	Beschreibung
Testfall-Nummer	00008
Testart	Rubiks-cube-tracker installieren
Befehlszeile ausführen	#bin/sh rubiks-cube-tracker –webcam 0
Testziel	Ein Kociemba string von einem rubiks cube bekommen
Testvoraussetzungen	• Rubiks-cube-tracker repos • Kamera
Testfalldaten	Ein Rubiks cube in verschidene Kompositionen
Erwartetes Verhalten	Ein kociemba string im Terminal bekommen
Testergebnis	**X** Bestanden ☐ Nicht Bestanden
Fehlerkategorie	☐ Leicht **X** Mittel ☐ Schwerwiegend
Bemerkung	 Kociemba string : FRFLUBDDFUUURRDDBBBBRLFFDRBLURUDLR

```
2023-01-11 21:54:29,683            __init__.py     INFO: Saved side
2023-01-11 21:54:36,656            __init__.py     INFO: Saved side
2023-01-11 21:54:42,229            __init__.py     INFO: Saved side
2023-01-11 21:54:47,856            __init__.py     INFO: Saved side
2023-01-11 21:54:57,791            __init__.py     INFO: Saved side
2023-01-11 21:55:05,842            __init__.py     INFO: Saved side
2023-01-11 21:55:05,844            __init__.py     INFO: rubiks-col
 --json --filename /tmp/webcam.json
Cube

          Gr Rd Gr
          OR Wh Bu
          Ye Ye Gr
 Rd Wh OR  Bu Bu Rd  Wh Wh Wh  OR Ye Ye
 Ye OR Gr  OR Gr Gr  Rd Rd Ye  Gr Bu OR
 Bu Gr Gr  Ye Rd Bu  Ye Bu Bu  Wh Bu Wh
          OR Wh Rd
          Wh Ye OR
          Rd Rd OR
2023-01-11 21:55:05,925            __init__.py     INFO: cd ~/rubik
olver/; ./usr/bin/rubiks-cube-solver.py --colormap '{"B": "Bu", "D"
Gr", "L": "OR", "R": "Rd", "U": "Wh"}' --state FRFLUBDDFUUURRDDBBBB
RLRULDLFBFFLDDFBLUBU
```

Tester Kunde	Tester Auftragnehmer	Datum 11.01.2022

1.9 Testfall 9: Ein Kociemba String Lösen :

Testfall	Beschreibung
Testfall-Nummer	00009
Testart	Rubiks-cube-tracker installieren
Befehlszeile ausführen	#bin/sh rubiks-cube-tracker –webcam 0
Testziel	Ein Kociemba string von einem rubiks cube bekommen
Testvoraussetzungen	• Rubiks-cubesolver repos • Ein Kociemba String
Testfalldaten	Ein Kociemba String
Erwartetes Verhalten	Schritte für die Lösung im Terminal zeigen

Testergeb-nis	**X** Bestanden ☐ Nicht Bestanden
Fehlerkate-gorie	☐ Leicht **X** Mittel ☐ Schwerwiegend
Bemerkung	

```
2023-01-11 21:55:56,783                __init__.py:3644     INFO: koci
          : U2 L' D B' U2 L' B U R2 F U L' U2 F2 R2 B2 U F2 U' D2 F2
2023-01-11 21:55:56,783                __init__.py:3645     INFO: koci
(reversed) : U F2 D2 U F2 U' B2 R2 F2 U2 L U' F' R2 U' B' L U2 B D' L
2023-01-11 21:55:56,784                __init__.py:1509     INFO:

3x3x3: solve 3x3x3, 22 steps, 22 total steps
============================================
       U U U
       U U U
       U U U

L L L  F F F  R R R  B B B
L L L  F F F  R R R  B B B
L L L  F F F  R R R  B B B

       D D D
       D D D
       D D D

2023-01-11 21:55:56,784                __init__.py:1509     INFO:
```

```
Solution: U2 L' D B' U2 L' B U R2 F U L' U2 F2 R2 B2 U F2 U' D2 F2 U'
2023-01-11 21:55:56,785                __init__.py:4710     INFO: 22 s
2023-01-11 21:55:56,785     rubiks-cube-solver.py:129     INFO: *****
************************************************************
2023-01-11 21:55:56,786     rubiks-cube-solver.py:130     INFO: See /
cube-NxNxN-solver/index.html for more detailed solve instructions
2023-01-11 21:55:56,786     rubiks-cube-solver.py:131     INFO: *****
************************************************************

2023-01-11 21:55:56,788     rubiks-cube-solver.py:162     INFO: rubik
er.py end
2023-01-11 21:55:56,789     rubiks-cube-solver.py:163     INFO: Memor
bytes
2023-01-11 21:55:56,789     rubiks-cube-solver.py:164     INFO: Time
.087457
2023-01-11 21:55:56,789     rubiks-cube-solver.py:165     INFO:
(venv) zorghost@zorghost-VirtualBox:~/rubiks-cube-NxNxN-solver$
```

Tester Kunde	Tester Auftragnehmer	Datum 11.01.2022

2 Testprotokoll

Testfall-Nr.	Datum	Status	Fehler-kategorie	Datum 2. Lauf	Status 2. Lauf
01	10.12.2022	nicht bestanden	mittel		
02	11.12.2022	Bestanden	mittel		
03	13.12.2022	Bestanden	mittel		
04	23.12.2022	Bestanden	mittel		
05	07.01.2023	nicht bestanden	mittel		
06	10.01.2023	bestanden	mittel		
07	11.01.2023	bestanden	mittel		
08	11.01.2023	bestanden	mittel		
09	11.01.2023	bestanden	mittel		
10					
11					

Aktueller Stand (?):
- Rot: nicht bestanden
- Grün: Bestanden

Anhang

A Fehlerkategorien

Für die Abnahme des Systems sind folgende Fehlerklassen definiert:

- **3 = Schwerwiegender Mangel** Produktivsetzung nicht möglich (nachhaltige Störung des Softwareablaufes mit daraus resultierender Funktionsuntüchtigkeit des Systems bzw. Störung von Systemteilen, die zur Störung aller Arbeitsabläufe beim Auftraggeber führt.)
- **2 = Mittlerer Mangel** Produktivsetzung möglich, aber mangelhafte Funktionen nicht nutzbar (durch eine Störung treten in Teilen der Programmabläufe erhebliche Störungen auf, sodass Teile der Software nicht verwendbar sind.)
- **1 = Leichter Mangel** Produktivsetzung durch Workaround mit vertretbarem Zusatzaufwand möglich (alle anderen als die in den vorstehenden Prioritätsgraden beschriebenen Störungsbilder)

B Qualitätskriterien nach ISO 9126

Gruppe	Q-Kriterium	
Funktionalität Sind alle im Pflichtenheft aufgeführten Kriterien vorhanden und ausführbar?	Angemessenheit	Merkmale von Software, die sich auf das Vorhandensein und die Eignung einer Menge von Funktionen für spezifizierte Aufgaben beziehen.
	Richtigkeit	Merkmale von Software, die sich beziehen auf das Liefern der richtigen oder vereinbarten Ergebnisse oder Wirkungen.
	Interoperabilität	Merkmale von Software, die sich auf ihre Eignung beziehen, mit vorgegebenen Systemen zusammenzuwirken.
	Ordnungsmäßigkeit	Merkmale von Software, die bewirken, dass die Software anwendungsspezifische Normen oder Vereinbarungen oder gesetzliche Bestimmungen oder ähnliche Vorschriften erfüllt.
	Sicherheit	Merkmale von Software, die sich auf ihre Eignung beziehen, unberechtigten Zugriff, sowohl versehentlich als auch vorsätzlich, auf Programme und Daten zu verhindern.
Zuverlässigkeit Zu welchem Grad erfüllt die Software dauerhaft und korrekt die geforderten Funktionen?	Reife	Merkmale von Software, die sich auf die Häufigkeit von Versagen durch Fehlzustände in der Software beziehen.
	Fehlertoleranz	Merkmale von Software, die sich auf ihre Eignung beziehen, ein spezifiziertes Leistungsniveau bei Software-Fehlern oder Nicht-Einhaltung ihrer spezifizierten Schnittstelle zu bewahren.
	Wiederherstellbarkeit	Merkmale von Software, die sich beziehen auf die Möglichkeit, bei einem Versagen ihr Leistungsniveau wiederherzustellen und die direkt betroffenen Daten wiederzugewinnen, und auf die dafür benötigte Zeit und den benötigten Aufwand.
Benutzbarkeit Wie schnell kann man den Umgang mit der Software lernen und wie leicht ist sie zu bedienen?	Verständlichkeit	Merkmale von Software, die sich auf den Aufwand für den Benutzer beziehen, das Konzept und die Anwendung zu verstehen.
	Erlernbarkeit	Merkmale von Software, die sich auf den Aufwand für den Benutzer beziehen, ihre Anwendung zu erlernen. (z.B. Ablaufsteuerung, Eingabe, Ausgabe)
	Bedienbarkeit	Merkmale von Software, die sich auf den Aufwand für den Benutzer bei der Bedienung und Ablaufsteuerung beziehen.
Effizienz Wie sind zeitliches Verhalten und Ressourcenverbrauch bei gegebenen Systemvoraussetzungen?	Zeitverhalten	Merkmale von Software, die sich beziehen auf die Antwort- und Verarbeitungszeiten und auf den Durchsatz bei der Ausführung ihrer Funktionen.
	Verbrauchsverhalten	Merkmale von Software, die sich darauf beziehen, wie viele Betriebsmittel bei der Erfüllung ihrer Funktionen benötigt werden und wie lange.
Änderbarkeit Mit welchem Zeit- und Arbeitsaufwand lassen sich Änderungen sowie Fehlererkennung und -behebung durchführen?	Analysierbarkeit	Merkmale von Software, die sich auf den Aufwand beziehen, der notwendig ist, um Mängel oder Ursachen von Versagen zu diagnostizieren oder um änderungsbedürftige Teile zu bestimmen.
	Modifizierbarkeit	Merkmale von Software, die sich auf den Aufwand beziehen, der zur Ausführung von Verbesserungen, zur Fehlerbeseitigung oder zur Anpassung an Umgebungsänderungen notwendig ist.
	Stabilität	Merkmale von Software, die sich auf das Risiko unerwarteter Wirkungen von Änderungen beziehen.
	Prüfbarkeit	Merkmale von Software, die sich auf den Aufwand beziehen, der zur Prüfung der geänderten Software notwendig ist.
Übertragbarkeit Mit welchem Aufwand lässt sich die Software an geänderte/ verbesserte Systembedingungen anpassen bzw. in neuen Systemen einsetzen?	Anpassbarkeit	Merkmale von Software, die sich auf die Möglichkeit beziehen, sie an verschiedene festgelegte Umgebungen anzupassen, wenn nur Schritte unternommen oder Mittel eingesetzt werden, die für diesen Zweck für die betrachtete Software vorgesehen sind.
	Installierbarkeit	Merkmale von Software, die sich auf den Aufwand beziehen, der zur Installation der Software in einer festgelegten Umgebung notwendig ist.
	Konformität	Merkmale von Software, die bewirken, dass die Software Normen oder Vereinbarungen zur Übertragbarkeit erfüllt.
	Austauschbarkeit	Merkmale von Software, die sich beziehen auf die Möglichkeit, diese anstelle einer anderen Software in der Umgebung jener Software zu verwenden und auf den dafür notwendigen Aufwand.

C Qualitätskriterien für Dokumente

Für die Erreichung des Projektzieles, das Produkt „Dokument“ zu erzeugen, dass den fachlichen und technischen Anforderungen des Auftraggebers entspricht, ergeben sich z.B. die folgenden Qualitätsmerkmale:

Merkmal	Erläuterung	Mindest-anfordrg.	Prüfmöglichkeit
Eindeutigkeit	Eignung von Dokumenten zur unmissverständlichen Vermittlung von Informationen für jeden Leser		Keine offenen Fragen zu den einzelnen Abschnitten (Prüfung durch Gruppeninspektion und Diskussion)
Lesbarkeit	Eignung von Dokumenten zur Entnahme der darin enthaltenen Informationen	ja	Prüfung durch Einsatz eines unbedarften Testlesers, Vorhandensein eines Glossars, Erläuterung von Fachbegriffen
Verständlichkeit	Eignung von Dokumenten zur erfolgreichen Vermittlung der darin enthaltenen Informationen an einen sachkundigen Leser	ja	Vorhandensein eines Glossars, Integration von Illustrationen, Diagrammen
Detaillierungsgrad	Vorhandensein der ausreichenden Beschreibung der fachlichen und technischen Einzelheiten im Dokument		Beschreibung der Sonder- und Ausnahmefälle, gleiche Behandlung (gleiche Detaillierung) aller Textabschnitte
Funktionale Vollständigkeit	Vorhandensein der für den Zweck der Dokumentation notwendigen und hinreichenden Information	ja	Einsatz des <KUNDE>Templates gewährleistet die Vollständigkeit an notwendigen Informationen, Beschreibung der Sonder- und Ausnahmefälle
Fehlerfreiheit	Nichtvorhandensein von sprachlichen Fehlern, die die Informationsaufnahme beeinträchtigen		Rechtschreib- und Grammatikprüfung
Widerspruchsfreiheit	Nichtvorhandensein von einander entgegenstehenden Aussagen im Dokument		Unnötige Redundanzen sollen vermieden werden, Dokument soll in sich konsistent sein
Aktualität	Übereinstimmung der Beschreibung der Situation in Dokument und Wirklichkeit		Gespräche mit dem Auftraggeber (Kundeninspektion, Workshops)
Funktionale Korrektheit	Nichtvorhandensein von funktionalen Fehlern, die den fachlichen und technischen Inhalt betreffen	ja	Wiedergabe der Anforderungen aus dem Vorgängerdokument
Normenkonformität	Erfüllung der für die Erstellung von Dokumenten geltenden Vorschriften und Normen		Einsatz des <KUNDE>Templates gewährleistet die formale Richtigkeit
Änderbarkeit	Eignung von Dokumenten zur Ermittlung aller von einer Änderung betroffenen Dokumententeile und zur Durchführung der Änderung		Einsatz des <KUNDE>Templates gewährleistet die formale Änderbarkeit, unnötige Redundanzen sollen vermieden werden