**Fallstudie**

**IPWA01-01**

**Aufgabenstellung 1**

**co2-Footprint**

von

**Kevin Faller**

Klosterstr. 30

77723 Gengenbach

Bearbeitungszeitraum: 1. November 2022 – 22. Juni 2023

GitHub <https://github.com/KevFaller>

Wissenschaftliche Kursleitung Prof. Dr. Marian Benner-Wickner

Tutor Paul Libbrecht

**Vorwort**

Dies ist eine Fallstudie zum Nachweis der Prüfungsleistung für das Modul IPWA02-01. Recht herzlich bedanken möchte ich mich bei Prof. Dr. Marian Banner-Wickner, sowie bei meinem Tutor Paul Libbrecht, die die Betreuung meiner Fallstudie übernommen haben.

Vielen Dank!

**Eidesstattliche Erklärung**

Hiermit versichere ich eidesstattlich, dass die vorliegende Fallstudie von mir selbstständig und ohne unerlaubte fremde Hilfe angefertigt worden ist, insbesondere, dass ich alle Stellen, die wörtlich oder annähernd wörtlich oder dem Gedanken nach aus Veröffentlichungen, unveröffentlichten Unterlagen und Gesprächen entnommen worden sind, als solche an den entsprechenden Stellen innerhalb der Arbeit durch Zitate kenntlich gemacht habe, wobei in den Zitaten jeweils der Umfang der entnommenen Originalzitate kenntlich gemacht wurde. Ich bin mir bewusst, dass eine falsche Versicherung rechtliche Folgen haben wird.



(Datum, Unterschrift)

20.07.2023

Diese Fallstudie ist urheberrechtlich geschützt, unbeschadet dessen wird folgenden Rechtsübertragungen zugestimmt:

• der Übertragung des Rechts zur Vervielfältigung der Bachelor-Thesis für Lehrzwecke an der IU-Internationale Hochschule (§ 16 UrhG),

• der Übertragung des Vortrags-, Aufführungs- und Vorführungs-rechts für Lehrzwecke durch Professoren der IU-Internationale Hochschule (§ 19 UrhG),

• der Übertragung des Rechts auf Wiedergabe durch Bild- oder Ton-

träger an die IU-Internationale Hochschule (§21 UrhG).

**Zusammenfassung**

Die vorliegende wissenschaftliche Arbeit basiert auf der Fallstudie mit dem Titel "Fallstudie IPWA01-01: co2-Footprint". Der Autor Kevin Faller, hat diese Fallstudie im Rahmen seines Studiums zum Softwareentwickler an der IU - Internationale Hochschule erstellt. Die Arbeit wurde von Prof. Dr. Marian Benner-Wickner betreut.

Das Ziel der Fallstudie war es, eine Webseite für eine fiktive Non-Profit-Organisation zu entwickeln, die sich mit dem Thema Klimawandel und co2-Emissionen beschäftigt. Die Webseite sollte einen Titel mit Logo, einen Header mit globaler Navigation, einen Content-Bereich und einen Footer mit rechtlichen Hinweisen enthalten. Die Webseite sollte responsive sein und sowohl auf Desktop-Computern als auch auf Smartphones gut lesbar sein. Der Content bestand aus einer Tabelle mit fiktiven co2-Werten, die nach Land und Unternehmen sortiert und gefiltert werden konnte. Zudem mussten die Eingabefelder gegen injizierten Code abgesichert werden.

Die verwendeten Technologien für die Implementierung der Webseite waren HTML, CSS und JavaScript. Zur Versionsverwaltung und Veröffentlichung des Quellcodes wurde ein GitHub-Repository genutzt. Visual Studio Code wurde als Entwicklungseditor verwendet und die Extension "Live Server" ermöglichte eine einfache visuelle Darstellung der Webseite während der Entwicklung.

Die Arbeit befasste sich auch mit den Grundlagen von HTML, CSS und JavaScript, sowie deren Bedeutung für die Frontend-Entwicklung. GitHub wurde als Versionsverwaltungssystem erklärt, und Visual Studio Code wurde als Quelltexteditor vorgestellt.

Die Fallstudie wurde erfolgreich umgesetzt und die Webseite erfüllte alle gestellten Anforderungen. Die Arbeit schließt mit einer Diskussion die verwendeten Technologien thematisiert, Templates, GitHub und der Sicherung der Eingabefelder. Ein Fazit rundet die Arbeit ab.

Das Literaturverzeichnis und das Quellcodeverzeichnis sind in der Arbeit ebenfalls enthalten.

Inhaltsverzeichnis

[1 Einleitung 6](#_Toc138578413)

[1.1 Motivation 6](#_Toc138578414)

[1.2 Zielsetzung 6](#_Toc138578415)

[1.3 Vorgaben 6](#_Toc138578416)

[2 Fallvorstellung 6](#_Toc138578417)

[3 Verwendete Technologien 7](#_Toc138578418)

[3.1 HTML und HTTP 7](#_Toc138578419)

[3.2 CSS 7](#_Toc138578420)

[3.3 JavaScript 7](#_Toc138578421)

[3.4 GitHub 7](#_Toc138578422)

[3.5 Visual Studio Code 8](#_Toc138578423)

[3.5.1 Live Server 8](#_Toc138578424)

[4 Implementierung 8](#_Toc138578425)

[4.1 GitHub Repository 8](#_Toc138578426)

[4.2 Templates 8](#_Toc138578427)

[4.2.1 Logo 9](#_Toc138578428)

[4.2.2 Head 9](#_Toc138578429)

[4.2.3 Body 9](#_Toc138578430)

[4.2.4 Footer 10](#_Toc138578431)

[4.3 Schriftkultur RTL - LTR 11](#_Toc138578432)

[4.4 co2 – Tabelle 12](#_Toc138578433)

[4.4.1 Tabelle generieren 12](#_Toc138578434)

[4.4.2 Tabelle filtern 13](#_Toc138578435)

[4.5 Webseite absichern (injizierter Code) 13](#_Toc138578436)

[5 Diskussion 13](#_Toc138578437)

[5.1 Technologie 14](#_Toc138578438)

[5.2 Templates 14](#_Toc138578439)

[5.3 GitHub 14](#_Toc138578440)

[5.4 Injizierter Code 14](#_Toc138578441)

[5.5 Generierung der Tabelle 14](#_Toc138578442)

[5.6 Tabelle Filtern 14](#_Toc138578443)

[6 Fazit 14](#_Toc138578444)

[7 Literaturverzeichnis 16](#_Toc138578445)

[8 Quellcodeverzeichnis 16](#_Toc138578446)

[9 Abbildungsverzeichnis 16](#_Toc138578447)

# 1 Einleitung

## Motivation

Diese Fallstudie wurde im Rahmen meines Studiums zum Softwareentwicklers bei der IU – Internationale Hochschule als Prüfungsleistung für das Modul IPWA02-01 erstellt. Zur Überprüfung meiner Leistung habe ich mir die Aufgabe 3 „Ghost Net Fishing“ ausgesucht.

## Zielsetzung

Diese wissenschaftliche Arbeit hat das Ziel, eine Webanwendung zur Meldung und Bergung von Geisternetzen zu entwickeln. Das Projekt wird im Web-Development-Team der Non-Profit-Organisation "Shea Shepherd" durchgeführt. Die Anwendung soll es meldenden Personen ermöglichen, anonym Geisternetze zu erfassen, während bergende Personen sich für die Bergung eines Netzes eintragen und den Status der noch zu bergenden Geisternetze einsehen können. Zudem sollen bergende Personen die Möglichkeit haben, erfolgreich geborgene Netze zu melden.

Die Arbeit wird gemäß den Anforderungen des Product Backlogs und der MoSCoW-Methode durchgeführt. Im ersten Schritt wird ein öffentliches Code-Repository, wie beispielsweise GitHub, für das Projekt eingerichtet. Anschließend wurden diese 5 aufgaben aus dem Product Backlog ausgewaehlt und Implementiert

1. MUST: Als meldende Person möchte ich Geisternetze (anonym) erfassen können.

2. MUST: Als bergende Person will ich mich für die Bergung eines Geisternetzes eintragen können.

3. MUST: Als bergende Person möchte ich sehen, welche Geisternetze noch zu bergen sind.

4. MUST: Als bergende Person möchte ich Geisternetze als geborgen melden können.

5. COULD: Als beliebige Person möchte ich Geisternetze als verschollen melden können.

## Vorgaben

Zu Beginn der Fallstudie werden momentan verfügbare Technogien untersucht, welche sich am besten zur Umsetzung eignen. Vorgegeben ist die Verwendung von HTML, JSP, CSS, JavaEE und JavaScript. Das Verwenden von weiteren JavaScript-Programmbibliotheken wie React.js oder Angula.js wurde freigestellt, ebenso die zu verwendende Entwicklungsumgebung.

# Fallvorstellung

Das Web-Development-Team der Non-Profit-Organisation "Shea Sepherd" hat sich zum Ziel gesetzt, eine Web-App zur Meldung und Bergung von Geisternetzen zu entwickeln. Geisternetze sind herrenlose Fischernetze, die im Meer treiben und eine ernsthafte Bedrohung für die marine Umwelt darstellen. Die Organisation möchte diese Gefahr mindern und das Bewusstsein für das Problem erhöhen.

Der Requirements Engineer hat bereits vor Beginn des Projekts intensive Gespräche mit verschiedenen Stakeholdern, sowohl innerhalb der Organisation als auch in der Fischereibranche, geführt, um die Anforderungen und Bedürfnisse zu verstehen. Das Fachkonzept wurde entwickelt und enthält wichtige Informationen über Geisternetze. Ein Geisternetz hat Eigenschaften wie den Standort (GPS-Koordinaten), eine geschätzte Größe und einen Status.

Der Status eines Geisternetzes kann verschiedene Ausprägungen haben: Gemeldet, Bergung bevorstehend, Geborgen oder Verschollen. Meldende Personen können Geisternetze anonym erfassen, während bergende Personen sich für die Bergung eines Netzes eintragen und den Status der noch zu bergenden Geisternetze einsehen können. Eine wichtige Einschränkung besteht darin, dass Geisternetze immer nur einer bergenden Person zugeordnet werden können, um unnötige Bergungsfahrten und Missverständnisse zu vermeiden. Bergende Personen können jedoch mehrere Geisternetze gleichzeitig bergen.

Das Projektteam hat die Anforderungen gemäß der MoSCoW-Methode priorisiert. Es gibt Must-have-Anforderungen, die zwingend umgesetzt werden müssen, wie das Erfassen von Geisternetzen, das Eintragen zur Bergung, das Anzeigen verfügbarer Geisternetze und das Melden von geborgenen Netzen. Zudem gibt es eine Could-have-Anforderung, die es beliebigen Personen ermöglicht, Geisternetze als verschollen zu melden.

Das Projekt ist agil konfiguriert, und du bist Teil eines interdisziplinären Teams von Entwickler:innen. Die Aufgabenstellung umfasst die Einrichtung eines öffentlichen Code-Repositories, die Auswahl von fünf Anforderungen aus dem Product Backlog für die prototypische Umsetzung im ersten Sprint, die Entwicklung eines Oberflächenprototypen mit Schwerpunkt auf persistenter Datenhaltung sowie die Dokumentation der Entwicklung der Webanwendung, einschließlich Design, Umsetzung, Datenbankstruktur und softwaretechnischer Architektur.

Der Product Owner erwartet einen funktionalen Prototypen mit einer pragmatischen Gestaltung der Nutzeroberfläche. Das Ergebnis muss keinen Design Award gewinnen, sondern vor allem die Anforderungen erfüllen und einen effektiven Beitrag zur Meldung und Bergung von Geisternetzen leisten.

# Verwendete Technologien

## HTML und HTTP

Die Auszeichnungssprache HTML ist neben der Stilsprache CSS und der Programmiersprache JavaScript eine der drei groben Sprachen, die für die Frontend-Entwicklung wichtig sind. (Ackermann, 2021, p. 47) *Das HyperText Transfer Protocoll* (HTTP), das ein Anwendungsprotokoll ist, definiert, wie Webserver und Webbrowser unterereinander kommunizieren. Das Protokoll selbst ist relativ einfach gehalten, um den Webserver möglichst wenig zu belasten. Die Kommunikation selbst setzt ein Transportprotokoll wie TCP/IP voraus. Eine Kommunikation über HTTP ist immer gleich aufgebaut (Herold, 2017, p. 478)

## JavaServerPages

JavaServer Pages (JSP) ist eine Java-Technologie, die zur Erstellung dynamischer Webseiten verwendet wird. Es ermöglicht die Integration von Java-Code direkt in den HTML-Code, wodurch eine nahtlose Verbindung zwischen dem Frontend und dem Backend hergestellt wird. JSP wird oft in Kombination mit JavaEE eingesetzt, da es die Erstellung komplexer, serverseitiger Anwendungen erleichtert.

Durch die Verwendung von JSP in JavaEE können Entwickler:innen die Leistungsfähigkeit der Java-Plattform voll ausschöpfen. Es bietet eine robuste Umgebung für die Entwicklung von unternehmenskritischen Anwendungen und unterstützt die Skalierbarkeit und Wiederverwendbarkeit von Code. Darüber hinaus erleichtert JSP die Trennung von Präsentationslogik und Geschäftslogik, was zu einer verbesserten Wartbarkeit und Erweiterbarkeit der Anwendung führt. Die Kombination von JSP und JavaEE bietet Entwickler:innen ein starkes Werkzeugset, um effiziente, sichere und zuverlässige Webanwendungen zu erstellen.

## Java Enterprise Edition

JavaEE (Java Enterprise Edition) ist eine umfangreiche Sammlung von Java-Technologien und APIs, die für die Entwicklung von skalierbaren, verteilten und unternehmenskritischen Anwendungen konzipiert sind. Es bietet eine robuste Plattform mit Funktionen wie Servlets, JavaServer Pages (JSP), Enterprise JavaBeans (EJB), Java Persistence API (JPA) und Java Message Service (JMS), die Entwickler:innen bei der Erstellung leistungsstarker und zuverlässiger Anwendungen unterstützen. JavaEE bietet eine standardisierte und stabile Entwicklungsumgebung, die bewährte Muster und bewährte Praktiken für die Unternehmensentwicklung nutzt. Es erleichtert die Entwicklung von Anwendungen, die hohe Anforderungen an Sicherheit, Skalierbarkeit und Zuverlässigkeit erfüllen müssen. JavaEE wird häufig in Unternehmen eingesetzt, um Enterprise-Anwendungen zu entwickeln, die die Anforderungen an komplexe Geschäftsprozesse und Datenmanagement erfüllen und gleichzeitig die Flexibilität und Wiederverwendbarkeit von Java nutzen.

## CSS

HTML beschreibt, was dargestellt werden soll, […]. Aber es beschreibt nicht, wie die Information dem Benutzer präsentiert werden soll. Über so genannte Cascading Style Sheets (CSS) kann dem Browser mitgeteilt werden, in welchen Schriftstil, -größe, -farbe, Hintergrund, Ausrichtung und Abstand der jeweilige Text darstellen soll. Teilweise kann auch noch das Verhalten bei Benutzerinteraktion modifiziert werden. (Herold, 2017, p. 479)

## Maven

Maven ist ein leistungsstarkes Build-Management-Tool für die Java-Entwicklung, das die Automatisierung des Build-Prozesses und die Verwaltung von Abhängigkeiten ermöglicht. Es bietet eine standardisierte Projektstruktur, in der Entwickler:innen ihre Projekte organisieren können, sowie eine zentrale Konfigurationsdatei (pom.xml), die Informationen über das Projekt, seine Abhängigkeiten und Build-Einstellungen enthält. Durch die Verwendung von Maven können Entwickler:innen effizientere Build-Prozesse erstellen, Abhängigkeiten automatisch verwalten und die Wiederverwendbarkeit von Code fördern.

## Docker

Docker ist eine Open-Source-Plattform, die es Entwickler:innen ermöglicht, Anwendungen in Containern zu verpacken und bereitzustellen. Container bieten eine konsistente und isolierte Umgebung, in der Anwendungen unabhängig von der zugrunde liegenden Infrastruktur ausgeführt werden können, was die Portabilität und Skalierbarkeit von Anwendungen erleichtert.

### Docker Desktop

Docker Desktop ist eine benutzerfreundliche Desktop-Anwendung, die es Entwickler:innen ermöglicht, Docker-Container auf ihren lokalen Systemen auszuführen und zu verwalten. Es bietet eine intuitive Benutzeroberfläche, um Container zu erstellen, zu starten, zu stoppen und zu überwachen, und ermöglicht somit eine einfache Entwicklung und Testung von Anwendungen in einer Containerumgebung.

### Docker-Compose

Docker Compose ist ein Tool, das es ermöglicht, mehrere Docker-Container als Teil einer Anwendung zu definieren und zu orchestrieren. Es bietet eine einfache Möglichkeit, komplexe Anwendungen mit verschiedenen Komponenten wie Datenbanken, DB-Management-Tools und Webservern zu konfigurieren und zu verwalten.

Docker Compose ermöglicht es, die gesamte Infrastruktur einer Anwendung in einer einzigen Konfigurationsdatei zu definieren, was die Wartung und Bereitstellung erleichtert. Durch die Verwendung von Docker Compose können Entwickler:innen die Container für die einzelnen Komponenten einer Anwendung mit nur einem Befehl erstellen, starten und stoppen.

Die Verwendung von Docker Compose ist besonders vorteilhaft in wissenschaftlichen Arbeiten, in denen komplexe Anwendungen mit mehreren Komponenten benötigt werden. Es vereinfacht die Einrichtung und den Betrieb von Entwicklungsumgebungen, indem es ermöglicht, alle erforderlichen Dienste und Tools, wie Datenbanken und Webserver, mit wenigen Schritten zu starten und zu verbinden. Dadurch können Entwickler:innen effizienter arbeiten und sich auf ihre eigentlichen Forschungs- und Entwicklungsaktivitäten konzentrieren, anstatt sich mit der manuellen Einrichtung und Konfiguration der einzelnen Komponenten auseinandersetzen zu müssen.

## JavaScript

Reine HTML-Dokumente erlauben nur sehr begrenzte Interaktion mit dem Benutzer, über Hyperlinks zu andere Dokumenten sowie über Formulare. Dabei steckt die ganze Logik hinter der Interaktion in der Implementierung des Servers. Schon Ende 1995 wurde erkannt, dass für eine bessere Bedienung eine echte Programmiersprache nötig ist, und mit JavaScript eine solche entwickelt. Trotz der Namensähnlichkeit hat JavaScript nichts mit der Sprache Java zu tun. (Herold, 2017, p. 480)

## GitHub

GitHub setzt sich aus zwei Worten zusammen, Git und Hub. Git ist eine Versionsverwaltungssoftware, entwickelt von dem Linuxerfinder Linus Torvalds. Hub wird ein Ort oder eine Logik bezeichnet Beispiele dafür sind Verteilzentren, Verkehrsknotenpunkte oder Güterverteilzentren. Das Besondere daran ist, dass man seinen Quellcode nicht nur verwalten, sondern auch anderen Menschen zur Verfügung stellen kann. GitHub steht so für Verwalten und Teilen.

## IntelliJ IDEA

IntelliJ IDE ist eine integrierte Entwicklungsumgebung (IDE), die von JetBrains entwickelt wurde und hauptsächlich für die Java-Programmierung verwendet wird. Sie bietet eine umfangreiche Palette an Funktionen und Werkzeugen zur effizienten Entwicklung, Fehlerbehebung und Codeverwaltung. Mit seiner intuitiven Benutzeroberfläche und Unterstützung für verschiedene Technologien erleichtert IntelliJ IDEA Entwickler:innen die Erstellung qualitativ hochwertiger Software.

# Implementierung

Beim Vorgehen nach der MoSCoW-Methode beginnt die Priorisierung mit der Identifizierung und Gewichtung der Must-have-Anforderungen. Diese Anforderungen sind von kritischer Bedeutung für den Erfolg des Projekts und müssen unbedingt erfüllt werden, um das definierte Ziel zu erreichen. Anschließend werden die Soll-, Kann- und Werden-Nicht-Anforderungen entsprechend ihrer relativen Wichtigkeit und Relevanz für das Projekt bewertet und priorisiert. Diese systematische Priorisierung ermöglicht es, Ressourcen und Aufmerksamkeit auf die Schlüsselkomponenten des Projekts zu konzentrieren und sicherzustellen, dass die entscheidenden Anforderungen erfüllt werden.

1. MUST: Als meldende Person möchte ich Geisternetze (anonym) erfassen können.

2. MUST: Als bergende Person will ich mich für die Bergung eines Geisternetzes eintragen können.

3. MUST: Als bergende Person möchte ich sehen, welche Geisternetze noch zu bergen sind.

4. MUST: Als bergende Person möchte ich Geisternetze als geborgen melden können.

5. COULD: Als beliebige Person möchte ich Geisternetze als verschollen melden können.

## Geisternetze (anonym) erfassen

Unter der URL http://localhost:8080/netz/netzBearbeiten besteht die Möglichkeit, Geisternetze zu melden oder den Status eines Geisternetzes zu ändern. Um ein Geisternetz zu melden, wählt die Person zunächst im Top-Down-Menü den Punkt "Melden" aus und gibt anschließend die GPS-Daten sowie die geschätzte Größe des Netzes an. Diese Informationen ermöglichen es, das Geisternetz zu lokalisieren und entsprechende Maßnahmen einzuleiten.



Abbildung

# Fazit

Im Rahmen der Fallstudie "CO²-Footprint" wurde eine öffentlich zugängliche Webseite für eine Non-Profit-Organisation entwickelt, die sich mit dem Thema Klimawandel und CO²-Emissionen beschäftigt. Das Ziel bestand darin, mehr Transparenz über die jährlichen CO²-Emissionen von Unternehmen und Ländern zu schaffen. Die Webseite wurde entsprechend der vorgegebenen Aufgabenstellung entworfen und umgesetzt.

Zur Umsetzung der Webseite wurden verschiedene Technologien eingesetzt. HTML wurde verwendet, um die grundlegende Struktur der Webseite zu definieren. CSS kam zum Einsatz, um das Design und das Erscheinungsbild der Webseite zu gestalten. JavaScript wurde verwendet, um interaktive Elemente und Funktionen auf der Webseite zu implementieren.

Ein öffentliches Code-Repository wurde in GitHub eingerichtet, um den Quellcode der Webseite zu verwalten und den Überprüfungsprozess zu ermöglichen. Die Entwicklungsumgebung Visual Studio Code wurde genutzt, um den Code zu schreiben und zu bearbeiten. Die Live Server Extension in Visual Studio Code ermöglichte eine einfache lokale Hostung der Webseite während der Entwicklungsphase.

Die Webseite erfüllt die gestellten Anforderungen der Fallstudie. Sie enthält einen Titel und ein Logo, einen Header mit globaler Navigation, einen Content-Bereich und einen Footer mit rechtlichen Hinweisen. Das Menü mit lokalen Links wurde entsprechend der Schriftkultur der Besucher rechts oder links dargestellt. Die Webseite ist responsiv und kann auf verschiedenen Geräten wie Desktop-Monitoren, Tablets und Smartphones gut gelesen werden. Eine Tabelle mit fiktiven CO²-Emissionsdaten wurde implementiert und ermöglicht das Sortieren und Filtern nach Land und Unternehmen. Alle Eingabefelder wurden abgesichert, um die Ausführung von injizierten Codes zu verhindern.

Die Implementierung der Webseite erfolgte unter Berücksichtigung bewährter Praktiken und Standards in der Webentwicklung. Der Quellcode wurde im GitHub-Repository abgelegt und dokumentiert.

Insgesamt wurde die Aufgabenstellung der Fallstudie erfolgreich umgesetzt. Die erstellte Webseite ermöglicht es der Non-Profit-Organisation, den CO²-Footprint von Unternehmen und Ländern transparent darzustellen und trägt somit zur Sensibilisierung für den Klimawandel bei. Durch die Verwendung geeigneter Technologien und die Beachtung von Sicherheitsaspekten wurde eine professionelle und benutzerfreundliche Webanwendung entwickelt.

# Literaturverzeichnis

Öggl, B., & Kofler, M. (2022). *Git - Projektverwaltung für Entwickler und DevOps-Teams.* Rheinwerk.

Ackermann, P. (2021). *Webentwicklung.* Rheinwerk Compting.

Herold, L. W. (2017). *Grundlagen der Informatik.* Pearson.

# Quellcodeverzeichnis

[Quellcode 1 head 9](#_Toc138422232)

[Quellcode 2 Menupunkt 1 10](#_Toc138422233)

[Quellcode 3 article 10](#_Toc138422234)

[Quellcode 4 Social Media 10](#_Toc138422235)

[Quellcode 5 Menu-RTL\LTR 11](#_Toc138422236)

[Quellcode 6 script-RTL\LTR 11](#_Toc138422237)

[Quellcode 7 Tabelle generieren 12](#_Toc138422238)

[Quellcode 8 injizierter Code 13](#_Toc138422239)

# Abbildungsverzeichnis

**Es konnten keine Einträge für ein Abbildungsverzeichnis gefunden werden.**