

Curriculum für

Certified Professional for  
Software Architecture (CPSA)<sup>®</sup>  
*Advanced Level*

**Modul  
GREEN**

**Green Software - Entwicklung ressourceneffizienter  
Anwendungen**

2024.1-rev0-DE-20240705



## Inhaltsverzeichnis

Verzeichnis der Lernziele .....	2
Einführung: Allgemeines zum iSAQB Advanced Level .....	3
Was vermittelt ein Advanced Level Modul? .....	3
Was können Absolventen des Advanced Level (CPSA-A)? .....	3
Voraussetzungen zur CPSA-A-Zertifizierung .....	3
Grundlegendes .....	4
Was vermittelt das Modul „GREEN“? .....	4
Struktur des Lehrplans und empfohlene zeitliche Aufteilung .....	4
Dauer, Didaktik und weitere Details .....	4
Voraussetzungen .....	5
Gliederung des Lehrplans .....	5
Ergänzende Informationen, Begriffe, Übersetzungen .....	5
1. Klimawandel, Chancen der Digitalisierung, Rolle der IT, Basiskonzepte .....	6
1.1. Begriffe und Konzepte .....	6
1.2. Lernziele .....	6
1.3. Referenzen .....	6
2. Prinzipien, Regulatorik und Auswirkungen auf Unternehmen .....	7
2.1. Begriffe und Konzepte .....	7
2.2. Lernziele .....	7
2.3. Referenzen .....	7
3. Qualität .....	8
3.1. Begriffe und Konzepte .....	8
3.2. Lernziele .....	8
3.3. Referenzen .....	8
4. Messen und Monitoring .....	9
4.1. Begriffe und Konzepte .....	9
4.2. Lernziele .....	9
4.3. Referenzen .....	10
5. Softwareentwicklung .....	11
5.1. Begriffe und Konzepte .....	11
5.2. Lernziele .....	11
5.3. Referenzen .....	11
6. Softwarearchitektur .....	12
6.1. Begriffe und Konzepte .....	12
6.2. Lernziele .....	12
6.3. Referenzen .....	12
7. Betrieb .....	13

7.1. Begriffe und Konzepte .....	13
7.2. Lernziele .....	13
7.3. Referenzen .....	14
8. Energieeffizienter Entwicklungsprozess .....	15
8.1. Begriffe und Konzepte .....	15
8.2. Lernziele .....	15
Referenzen .....	16

© (Copyright), International Software Architecture Qualification Board e. V. (iSAQB® e. V.) 2023

Die Nutzung des Lehrplans ist nur unter den nachfolgenden Voraussetzungen erlaubt:

1. Sie möchten das Zertifikat zum CPSA Certified Professional for Software Architecture Foundation Level® oder CPSA Certified Professional for Software Architecture Advanced Level® erwerben. Für den Erwerb des Zertifikats ist es gestattet, die Text-Dokumente und/oder Lehrpläne zu nutzen, indem eine Arbeitskopie für den eigenen Rechner erstellt wird. Soll eine darüber hinausgehende Nutzung der Dokumente und/oder Lehrpläne erfolgen, zum Beispiel zur Weiterverbreitung an Dritte, Werbung etc., bitte unter [info@isaqb.org](mailto:info@isaqb.org) nachfragen. Es müsste dann ein eigener Lizenzvertrag geschlossen werden.
2. Sind Sie Trainer oder Trainingsprovider, ist die Nutzung der Dokumente und/oder Lehrpläne nach Erwerb einer Nutzungslizenz möglich. Hierzu bitte unter [info@isaqb.org](mailto:info@isaqb.org) nachfragen. Lizenzverträge, die alles umfassend regeln, sind vorhanden.
3. Falls Sie weder unter die Kategorie 1. noch unter die Kategorie 2. fallen, aber dennoch die Dokumente und/oder Lehrpläne nutzen möchten, nehmen Sie bitte ebenfalls Kontakt unter [info@isaqb.org](mailto:info@isaqb.org) zum iSAQB e. V. auf. Sie werden dort über die Möglichkeit des Erwerbs entsprechender Lizenzen im Rahmen der vorhandenen Lizenzverträge informiert und können die gewünschten Nutzungsgenehmigungen erhalten.

#### Wichtiger Hinweis

**Grundsätzlich weisen wir darauf hin, dass dieser Lehrplan urheberrechtlich geschützt ist. Alle Rechte an diesen Copyrights stehen ausschließlich dem International Software Architecture Qualification Board e. V. (iSAQB® e. V.) zu.**

Die Abkürzung "e. V." ist Teil des offiziellen Namens des iSAQB und steht für "eingetragener Verein", der seinen Status als juristische Person nach deutschem Recht beschreibt. Der Einfachheit halber wird iSAQB e. V. im Folgenden ohne die Verwendung dieser Abkürzung als iSAQB bezeichnet.

## Verzeichnis der Lernziele

- LZ 1-1: Globale Herausforderungen des Klimawandels kennen und beachten
- LZ 1-2: Möglichkeiten zur Einsparung von Energie durch Digitalisierung kennen
- LZ 1-3: Energieverbrauch von IT einordnen, beziffern und deren Treiber kennen
- LZ 1-4: Handlungsfelder für das Einsparen von CO<sub>2</sub> kennen
- LZ 2-1: Regulatorik und deren Anforderungen kennen
- LZ 2-2: Greenhouse Gas Protocol kennen und anwenden
- LZ 2-3: Energieeffizienz als Handlungsfeld im Unternehmen platzieren
- LZ 2-4: Kennen der Stakeholder im Kontext von Green IT und deren Priorisierungen
- LZ 3-1: Qualitätsmodelle und deren Bezug zu Energieeffizienz kennen
- LZ 3-2: Umgang mit den Wechselwirkungen zwischen Qualitätsattributen und Energieeffizienz
- LZ 4-1: Ressourcen in Relation zur erbrachten Leistung bringen
- LZ 4-2: Messverfahren und Messwerkzeuge kennen und einsetzen
- LZ 4-3: Messmethodiken kennen und anwenden
- LZ 4-4: Integrationsmöglichkeiten zum Messen der Energieeffizienz im Entwicklungsprozess kennen
- LZ 5-1: Möglichkeiten zur Steigerung der Energieeffizienz in der Softwareentwicklung kennen
- LZ 5-2: Verfahren für energieeffizientes Datenhandling kennen und anwenden
- LZ 5-3: Effekte und Möglichkeiten von Optimierungen kennen
- LZ 6-1: Architekturstile und deren Bezug zu Energieeffizienz kennen und verstehen
- LZ 6-2: Kommunikation und deren Auswirkungen auf Energieeffizienz kennen
- LZ 6-3: Datenbanken und deren Eigenschaften in Bezug auf Energieeffizienz kennen
- LZ 6-4: Spezifische Green IT Pattern kennen und anwenden
- LZ 6-5: Web Sustainability Guidelines kennen und einsetzen
- LZ 7-1: Energieeffizienz von Rechenzentren und Hardware beziffern
- LZ 7-2: Cloud Service Modelle kennen und bezüglich Energieeffizienz beurteilen
- LZ 7-3: Cloud Deployment Modelle kennen und bezüglich Energieeffizienz beurteilen
- LZ 7-4: Anbieter nach ökologisch-nachhaltigen Aspekten bewerten und auswählen
- LZ 7-5: Betriebliche Antipatterns kennen
- LZ 7-6: CO<sub>2</sub>-Optimierung in der Cloud vornehmen
- LZ 8-1: CI/CD-Strategien und deren Ressourcenbedarf kennen
- LZ 8-2: Optimierung der CI/CD Prozesse durchführen
- LZ 8-3: Optimierung der Infrastruktur vornehmen
- LZ 8-4: Teststrategie optimieren

## Einführung: Allgemeines zum iSAQB Advanced Level

### Was vermittelt ein Advanced Level Modul?

Das Modul kann unabhängig von einer CPSA-F-Zertifizierung besucht werden.

- Der iSAQB Advanced Level bietet eine modulare Ausbildung in drei Kompetenzbereichen mit flexibel gestaltbaren Ausbildungswegen. Er berücksichtigt individuelle Neigungen und Schwerpunkte.
- Die Zertifizierung erfolgt als Hausarbeit. Die Bewertung und mündliche Prüfung wird durch vom iSAQB benannte Expert:innen vorgenommen.

### Was können Absolventen des Advanced Level (CPSA-A)?

CPSA-A-Absolventen können:

- eigenständig und methodisch fundiert mittlere bis große IT-Systeme entwerfen
- in IT-Systemen mittlerer bis hoher Kritikalität technische und inhaltliche Verantwortung übernehmen
- Maßnahmen zur Erreichung von Qualitätsanforderungen konzeptionieren, entwerfen und dokumentieren sowie Entwicklungsteams bei der Umsetzung dieser Maßnahmen begleiten
- architekturelevante Kommunikation in mittleren bis großen Entwicklungsteams steuern und durchführen

### Voraussetzungen zur CPSA-A-Zertifizierung

- erfolgreiche Ausbildung und Zertifizierung zum Certified Professional for Software Architecture, Foundation Level® (CPSA-F)
- mindestens drei Jahre Vollzeit-Berufserfahrung in der IT-Branche; dabei Mitarbeit an Entwurf und Entwicklung von mindestens zwei unterschiedlichen IT-Systemen
  - Ausnahmen sind auf Antrag zulässig (etwa: Mitarbeit in Open-Source-Projekten)
- Aus- und Weiterbildung im Rahmen von iSAQB-Advanced-Level-Schulungen im Umfang von mindestens 70 Credit Points aus mindestens drei unterschiedlichen Kompetenzbereichen
- erfolgreiche Bearbeitung der CPSA-A-Zertifizierungsprüfung



## Grundlegendes

### Was vermittelt das Modul „GREEN“?

Galt die IT lange Zeit als Lösung vieler Probleme, die in Zusammenhang mit dem Klimawandel stehen, ist sie mittlerweile selbst Gegenstand von Optimierungsbetrachtungen geworden. Ineffiziente Programmierung, oft hervorgerufen durch die Notwendigkeit eines schnellen Time-to-market, wurde vielfach durch immer schnellere Hardware oder mehr Ressourcen in der Cloud kompensiert. Diesen Weg gilt es zu verlassen. Die CO<sub>2</sub>-Emissionen, die durch Software entstehen, müssen konsequent reduziert werden.

Im Modul GREEN lernen die Teilnehmenden, das Thema Green Software ganzheitlich zu betrachten. Dies beginnt mit der Betrachtung der Rolle der IT beim Aufhalten des Klimawandels, einer Einführung in die aktuelle Regulatorik, die Sicht auf die Anforderungen verschiedener Stakeholder und der Benennung von Handlungsfeldern in Unternehmen. Von dort geht es über das Messen und das Monitoring von CO<sub>2</sub>-Emissionen bzw. Energieverbrauch hin zum Kernthema der Softwareentwicklung. In dieses zentrale Themengebiet fällt die Energieeffizienz verschiedener Softwarearchitekturen sowie grundlegender verwendeter Konzepte, energieeffizientes Datenhandling, optimierte Algorithmen aber auch der Einfluss und das Management von Qualitätsanforderungen in Bezug auf Energieeffizienz. Ein weiterer wichtiger Bestandteil ist das Thema Cloud, sowohl bezüglich Auswahl des Providers als auch den Möglichkeiten zum CO<sub>2</sub>-armen Betrieb darin. Zuletzt finden Optionen zur Verbesserung der Energieeffizienz im Entwicklungsprozess Betrachtung.

Am Ende des Moduls GREEN kennen die Teilnehmenden die wesentlichen Hebel, um CO<sub>2</sub> in der IT zu reduzieren. Sie können einschätzen, welche Auswirkungen die Softwarearchitektur hat und wie Architekturmuster- und -stile wirken. Sie kennen Fallstricke beim Datenhandling und können einschätzen, welche Auswirkungen die Auswahl von technischen Komponenten haben. In Bezug auf die Cloud wissen die Teilnehmenden die dort angebotenen Möglichkeiten im Sinne der Energieeffizienz einzusetzen.

### Struktur des Lehrplans und empfohlene zeitliche Aufteilung

Inhalt	Empfohlene Mindestdauer (min)
1. Klimawandel, Chancen der Digitalisierung, Rolle der IT, Basiskonzepte	60
2. Prinzipien, Regulatorik und Auswirkungen auf Unternehmen	60
3. Qualität	60
4. Messen und Monitoring	150
5. Softwareentwicklung	75
6. Softwarearchitektur	150
7. Betrieb	105
8. Energieeffizienter Entwicklungsprozess	60
Summe	720 (12h)

### Dauer, Didaktik und weitere Details

Die unten genannten Zeiten sind Empfehlungen. Die Dauer einer Schulung zum Modul GREEN sollte mindestens 2 Tage betragen, kann aber länger sein. Anbieter können sich durch Dauer, Didaktik, Art und

Aufbau der Übungen sowie der detaillierten Kursgliederung voneinander unterscheiden. Insbesondere die Art der Beispiele und Übungen lässt der Lehrplan komplett offen.

Lizenzierte Schulungen zu GREEN tragen zur Zulassung zur abschließenden Advanced-Level-Zertifizierungsprüfung (folgende Credit Points) bei:

Methodische Kompetenz:	10 Punkte
Technische Kompetenz:	10 Punkte
Kommunikative Kompetenz:	0 Punkte

## Voraussetzungen

Teilnehmerinnen und Teilnehmer **sollten** folgende Kenntnisse und/oder Erfahrung mitbringen:

- Praktische Erfahrung in Entwurf und Entwicklung kleiner bis mittelgroßer Softwaresysteme
- Kenntnisse über den Lebenszyklus von Softwaresystemen
- Umgang mit Qualitätsanforderungen
- Praktische Erfahrung beim Monitoring von Softwaresystemen
- Praktische Programmiererfahrung

**Hilfreich** für das Verständnis einiger Konzepte sind darüber hinaus:

- Erste praktische Erfahrungen mit Performance Engineering
- Erste praktische Erfahrung mit gängigen Cloud-Anbietern
- Erste praktische Erfahrung mit CI/CD-Pipelines

## Gliederung des Lehrplans

Die einzelnen Abschnitte des Lehrplans sind gemäß folgender Gliederung beschrieben:

- **Begriffe/Konzepte:** Wesentliche Kernbegriffe dieses Themas.
- **Unterrichts-/Übungszeit:** Legt die Unterrichts- und Übungszeit fest, die für dieses Thema bzw. dessen Übung in einer akkreditierten Schulung mindestens aufgewendet werden muss.
- **Lernziele:** Beschreibt die zu vermittelnden Inhalte inklusive ihrer Kernbegriffe und -konzepte.

Dieser Abschnitt skizziert damit auch die zu erwerbenden Kenntnisse in entsprechenden Schulungen.

## Ergänzende Informationen, Begriffe, Übersetzungen

Soweit für das Verständnis des Lehrplans erforderlich, haben wir Fachbegriffe ins [iSAQB-Glossar](#) aufgenommen, definiert und bei Bedarf durch die Übersetzungen der Originalliteratur ergänzt.



# 1. Klimawandel, Chancen der Digitalisierung, Rolle der IT, Basiskonzepte

Dauer: 60 Min.	Übungszeit: keine
----------------	-------------------

## 1.1. Begriffe und Konzepte

Klimawandel, Verbrauch von IT, Einsparungen durch IT, Einsparungen in IT, Nachhaltigkeit, CO<sub>2</sub>, CO<sub>2</sub>-Äquivalente, CO<sub>2</sub>-Fußabdruck, CO<sub>2</sub>-Handabdruck, negative externe Effekte, Watt, Joule, Effizienz, Effektivität

## 1.2. Lernziele

### LZ 1-1: Globale Herausforderungen des Klimawandels kennen und beachten

Die Teilnehmenden kennen die globalen Herausforderungen des Klimawandels. Sie kennen den steigenden Energie- und Ressourcenbedarf der IT und deren wesentliche Treiber. Sie können erklären, wieso sich auch die IT mit dem Thema Ressourcen- und Energieeffizienz beschäftigen muss.

### LZ 1-2: Möglichkeiten zur Einsparung von Energie durch Digitalisierung kennen

Die Teilnehmenden kennen die Möglichkeiten zur Einsparung von Energie und CO<sub>2</sub> durch die Digitalisierung. Sie kennen die Sektoren, die besonders von Digitalisierung profitieren. Der Unterschied zwischen Einsparungen in der IT und durch die IT ist bekannt.

### LZ 1-3: Energieverbrauch von IT einordnen, beziffern und deren Treiber kennen

Die Teilnehmenden können den Energieverbrauch der Informationstechnik beziffern, können diesen in Relation zu anderen Sektoren einordnen und die wesentlichen Treiber benennen. Sie kennen Trends und Entwicklungen in der IT und zentrale Stellschrauben.

### LZ 1-4: Handlungsfelder für das Einsparen von CO<sub>2</sub> kennen

Die Teilnehmenden kennen die Handlungsfelder zum Einsparen von CO<sub>2</sub>:

- Energieeffizienz - Verbrauche so wenig Energie wie möglich.
- Hardwareeffizienz - Beanspruche so wenig Hardware wie möglich.
- CO<sub>2</sub>-Emissionseffizienz - Verbrauche Energie möglichst dann und dort, wo sie am "grünsten" erzeugt wird.

## 1.3. Referenzen

[bitkom 2021], [Brundtland 1987]

## 2. Prinzipien, Regulatorik und Auswirkungen auf Unternehmen

Dauer: 60 Min.	Übungszeit: keine
----------------	-------------------

### 2.1. Begriffe und Konzepte

Energieeffizienz, Zertifizierungen (z. B. Blauer Engel, TCO), Regulatorische Anforderungen (z. B. CSRD, ESD), Greenhouse Gas Protocol (GHG Protocol)

### 2.2. Lernziele

#### LZ 2-1: Regulatorik und deren Anforderungen kennen

Die Teilnehmenden kennen die Regulatorik und die darin enthaltenen Anforderungen zur Einsparung von CO<sub>2</sub>. Sie kennen die vorhandenen Zertifizierungen, beispielsweise den Blauen Engel, und können diese adäquat in ihrer Arbeit einsetzen. Die Teilnehmenden kennen Standards wie ESG (Environmental Social Governance) und CSRD (Corporate Sustainability Reporting Directive) und deren Auswirkungen und Konsequenzen für Unternehmen.

#### LZ 2-2: Greenhouse Gas Protocol kennen und anwenden

Die Teilnehmenden kennen das Greenhouse Gas Protocol (GHP). Sie sind in der Lage die Scopes zu erklären und Emissionen den Scopes zuzuordnen. Sie können erklären, welche Scopes beim Betrieb von Software in der Cloud oder On Premise betroffen sind. Die Teilnehmenden kennen die ISO 14064-1 als Ableitung des Greenhouse Gas Protocols.

#### LZ 2-3: Energieeffizienz als Handlungsfeld im Unternehmen platzieren

Die Teilnehmenden kennen die Motivation, um Green Software im Unternehmen zu platzieren und Argumentationen und die Adressaten für eine erfolgreiche Verankerung im Unternehmen. Sie können sinnvolle Ziele für die Reduktion von Treibhausgasen für Unternehmen setzen. Die Teilnehmenden können erklären, welche Anforderungen Dritte entlang der Wertschöpfungskette haben und wie sich diese auf das eigene Unternehmen auswirken.

#### LZ 2-4: Kennen der Stakeholder im Kontext von Green IT und deren Priorisierungen

Die Teilnehmenden kennen die Anforderungen der verschiedenen Stakeholder an energieeffiziente Software und die daraus resultierenden Handlungsfelder. Sie kennen die Bereiche, bei denen die größten Hebel für die Reduktion von Treibhausgasen bestehen und können Maßnahmen entsprechend priorisieren. Die Teilnehmenden können unterschiedliche Softwaresysteme klassifizieren und Aufwand und Nutzen einer Optimierung erkennen.

### 2.3. Referenzen

[\[Clements et al. 2003\]](#)

### 3. Qualität

Dauer: 45 Min.	Übungszeit: 15 Min.
----------------	---------------------

#### 3.1. Begriffe und Konzepte

Qualitätsmodell, Qualitätsszenarien, Qualitätsziele, ISO 25010

#### 3.2. Lernziele

##### LZ 3-1: Qualitätsmodelle und deren Bezug zu Energieeffizienz kennen

Die Teilnehmenden kennen unterschiedliche Qualitätsmodelle und können Qualitätsattribute zum Thema Energieeffizienz beschreiben und in die Qualitätsmodelle einordnen. Darüber hinaus sind die Teilnehmenden in der Lage Qualitätsszenarien für Qualitätsattribute im Bereich Energieeffizienz zu formulieren. Abhängig vom Einsatzzweck und dem Mengengerüst der Anwendung können sie bewusst darüber entscheiden, ob die Qualitätskriterien nur für die Anwendung oder für die Optimierung des Entwicklungsprozesses angewendet werden.

##### LZ 3-2: Umgang mit den Wechselwirkungen zwischen Qualitätsattributen und Energieeffizienz

Die Teilnehmenden kennen die positiven und negativen Wechselwirkungen zwischen Qualitätsattributen im Bereich der Energieeffizienz mit anderen Qualitätsattributen. Auf Basis dieser Erkenntnisse können sie Qualitätsanforderungen erkennen, die miteinander in Konflikt stehen und Lösungsalternativen erarbeiten, um Konflikte zu abschwächen. Insbesondere werden die Qualitätsmerkmale Elastizität und Skalierbarkeit, Modularität bzw. Resilienz detaillierter diskutiert, so dass die Teilnehmenden den Einfluss auf Energieeffizienz bewerten können.

#### 3.3. Referenzen

[\[arc42\]](#), [\[ISO 25010\]](#)

## 4. Messen und Monitoring

Dauer: 90 Min	Übungszeit: 60 min
---------------	--------------------

### 4.1. Begriffe und Konzepte

Software Carbon Intensity (SCI), Business Metriken, Proxy-Metriken, Energieeffizienz, Kategorien von Messverfahren, Messwerkzeuge

### 4.2. Lernziele

#### LZ 4-1: Ressourcen in Relation zur erbrachten Leistung bringen

Die Teilnehmenden sind in der Lage, Ressourcen in Relation zur erbrachten Leistung zu bringen (Business Metrik). Sie kennen als Metrik den ISO-Standard Software Carbon Intensity (SCI) und verstehen, wie dieser als Business Metrik genutzt werden kann.

#### LZ 4-2: Messverfahren und Messwerkzeuge kennen und einsetzen

Die Teilnehmenden kennen verschiedene Kategorien von Messverfahren und deren Einsatzgebiete. Sie verstehen, wie man Werkzeuge der verschiedenen Kategorien für eine Analyse kombinieren kann. Kategorien sind beispielsweise:

- Hardware-Messwerkzeuge. Messen den Verbrauch direkt an der Hardware.
- Fullstack-Messwerkzeuge. Messen alle Komponenten einer Anwendung wie Verbräuche pro Docker-Container, Netzwerk.
- Programmiersprachenspezifische Messwerkzeuge. Liefern "Whitebox"-Messungen innerhalb einer Anwendung.
- Messwerkzeuge für Webanwendungen. Geben Auskunft u. a. über die Netzwerklast, die durch einen Seitenaufruf erzeugt wird.
- Carbon Footprint Tools der Cloudanbieter. In Cloudumgebungen kann der Energieverbrauch nicht direkt gemessen werden. Stattdessen ist man auf vom Anbieter bereitgestellte Tools oder andere Schätzmethode angewiesen. Die Teilnehmenden kennen die Funktionalität und die Einschränkungen dieser Tools (Ungenauigkeit, relative Angaben, nötige Berechtigungen in der Cloud-Console) wie auch die Herausforderungen, Managed Services mit reinen VMs zu vergleichen.

#### LZ 4-3: Messmethodiken kennen und anwenden

Die Teilnehmenden kennen das methodische Vorgehen beim Messen des Energieverbrauchs, bzw. der CO<sub>2</sub>-Intensität. Sie verstehen, dass die Bewertung / Messung des absoluten Verbrauchs i. d. R. (gerade in der Cloud) nicht sehr exakt möglich ist, sondern wie man stattdessen Trends analysiert. Sie kennen Proxy-Metriken für die CO<sub>2</sub>-Intensität.

Die Teilnehmenden wissen, wie sie sinnvolle Szenarien zum Messen der Energieeffizienz aufsetzen.

- Sie kennen Kriterien zum Gestalten der Szenarien (z. B. gängige Abläufe, Priorisierung nach Nutzungshäufigkeit).
- Sie wissen, an welchen Schnittstellen sie abhängig vom Ziel der Messung ansetzen (Ende-zu-Ende-Szenarien, an APIs ansetzen, etc.).
- Sie kennen Werkzeuge zum Erstellen solcher Szenarien.

- Sie können adäquat mit Messungenauigkeiten umgehen (z. B. Szenarien mehrfach ausführen, Abweichungen interpretieren).

Die Teilnehmenden verstehen, dass das Vorgehen der Energieeffizienzmessung verwandt ist mit dem Vorgehen bei Performanz-Messungen. Die Teilnehmenden verstehen, wie man systematisch vorgeht, um Verbrauchshotspots einzugrenzen.

#### **LZ 4-4: Integrationsmöglichkeiten zum Messen der Energieeffizienz im Entwicklungsprozess kennen**

Die Teilnehmenden wissen, wie man die Energieeffizienzmessungen in den Entwicklungsprozess integriert. Sie kennen Möglichkeiten, die Messungen in die CI/CD-Pipeline zu integrieren. Sie verstehen, wie sie eine gute Balance für die Häufigkeit der Messungen finden - einerseits sollen (negative) Trends schnell erkannt werden, andererseits verbraucht das Messen selbst ebenfalls Energie.

Die Teilnehmenden verstehen, dass die Messungen möglichst einfach und leichtgewichtig für die Entwickler:innen zu benutzen sein müssen, um die Akzeptanz zu erhöhen. Sie verstehen, dass die Energieeffizienzmessungen konkurrieren mit einem schnellen Feedback aus dem Build. Sie verstehen, dass folgende Punkte für die Akzeptanz ebenfalls wichtig sind:

- Die Erstellung der Szenarien für die Messung muss einfach sein. Es ist von Vorteil, dafür Werkzeuge zu verwenden, die im Projekt bereits etabliert sind (z. B. Frameworks zum Erstellen von Ende-zu-Ende-Tests)
- Die Darstellung der Messergebnisse muss übersichtlich, gut und schnell verständlich sein. Insbesondere müssen die wichtigsten Ergebnisse wie Trends oder signifikante Abweichungen im Verbrauch direkt erkennbar sein.

### **4.3. Referenzen**

[Greenhouse Gas Protocol], [SCI], [Greenframe], [Green Metrics Tool], [Cloud Carbon Footprint], [JoularJX], [Code Carbon]

## 5. Softwareentwicklung

Dauer: 75 Min.	Übungszeit: keine
----------------	-------------------

### 5.1. Begriffe und Konzepte

Programmiersprachen, Ahead-of-Time und Just-in-Time Compilierung, Bytecode, Maschine Code, Garbage-Collection, Datenstrukturen, Algorithmen, Rechnerische Komplexität, Caching, Datenbanksysteme, Datenmodelle

### 5.2. Lernziele

#### LZ 5-1: Möglichkeiten zur Steigerung der Energieeffizienz in der Softwareentwicklung kennen

Die Teilnehmenden kennen Vorgehensweisen und Grenzen bei der Optimierung von Programmen. Sie können sinnvolle Ansatzpunkte von wenig lohnenden Zielen unterscheiden. Die Teilnehmenden kennen die Energieeffizienz verschiedener Programmiersprachen und können einschätzen, für welche Einsatzgebiete diese geeignet sind. Sie können Laufzeitverhalten und Entwicklungsgeschwindigkeit von Programmiersprachen bewerten und basierend auf den Anforderungen die richtige Wahl treffen. Darüber hinaus verstehen sie die Unterschiede zwischen Ahead-of-Time und Just-in-Time Kompilierung, von Bytecode und Maschine Code sowie die Zielkonflikte von Garbage-Collectoren.

#### LZ 5-2: Verfahren für energieeffizientes Datenhandling kennen und anwenden

Die Teilnehmenden kennen Verfahren, um mit großen Datenmengen umzugehen. Sie können Datenstrukturen und Operationen auf diesen Datenstrukturen bezüglich Energieeffizienz einschätzen und die richtige Auswahl bezogen auf die gegebenen Anforderungen vornehmen. Die Teilnehmenden kennen Fallstricke beim Einsatz von Datenbanksystemen und können diese vermeiden. Sie sind in der Lage unterschiedliche Ansätze für die Abbildung von Beziehungen zwischen Entitäten auf Basis der fachlichen Anforderungen und der Qualitätsanforderungen zu bewerten und optimal auszuwählen.

#### LZ 5-3: Effekte und Möglichkeiten von Optimierungen kennen

Die Teilnehmenden kennen das Prinzip der rechnerischen Komplexität und können Algorithmen bezüglich ihrer Laufzeit- und Energieeffizienz bewerten und auswählen. Sie können Caches in Verbindung mit Energieeffizienz einsetzen, kennen deren Funktionsweise (z. B. Cache-Ersetzung) sowie die Vor- und Nachteile. Sie kennen Möglichkeiten und Grenzen von Werkzeugen zum Aufdecken von Energy Code Smells.

### 5.3. Referenzen

[\[Pereira et al.\]](#)

## 6. Softwarearchitektur

Dauer: 90 Min.	Übungszeit: 60 Min.
----------------	---------------------

### 6.1. Begriffe und Konzepte

Architekturstile, Monolith, Microservices, Serverless, Inter-Prozess-Kommunikation, Datenbanken, Green IT Pattern

### 6.2. Lernziele

#### LZ 6-1: Architekturstile und deren Bezug zu Energieeffizienz kennen und verstehen

Die Teilnehmenden können unterschiedliche Architekturstile wie beispielsweise verteilte Systeme, strukturierte Monolithen oder Serverless und deren Auswirkungen auf die Energieeffizienz einschätzen. Darüber hinaus sind sie in der Lage den Unterschied zwischen zustandslosen und zustandsbehafteten Komponenten hinsichtlich des Energiebedarfs einzuschätzen. Die Prinzipien von cloud-native Architekturen und ihre Auswirkungen auf die Energieeffizienz sind den Teilnehmenden bekannt.

#### LZ 6-2: Kommunikation und deren Auswirkungen auf Energieeffizienz kennen

Die Teilnehmenden kennen unterschiedliche Arten der Kommunikation (synchron versus asynchron) zwischen Bausteinen in Bezug auf die Bereitstellung von Daten sowie der verwendeten Formate und Protokolle (textbasierte versus binäre Protokolle) hinsichtlich der Energieeffizienz einschätzen. Sie sind in der Lage den Einfluss von Datenkompression auf das übertragene Datenvolumen zu bewerten und wissen, dass die Energieeffizienz durch eine Reduzierung von Aufrufhäufigkeit und des Datenvolumens positiv beeinflusst wird.

#### LZ 6-3: Datenbanken und deren Eigenschaften in Bezug auf Energieeffizienz kennen

Die Teilnehmenden kennen unterschiedliche Arten von Datenbankmodellen (relational, NoSQL, etc.) und unterschiedliche Datenbanksysteme (PostgreSQL, DB2, Oracle, etc.) und können die richtige Auswahl unter dem Aspekt der Energieeffizienz treffen. Sie sind in der Lage ein geeignetes Service-Modell (self-hosting versus cloud-based managed service) für ihre Datenhaltung auszuwählen.

#### LZ 6-4: Spezifische Green IT Pattern kennen und anwenden

Die Teilnehmenden kennen generell Muster wie beispielsweise die der Green Software Foundation, um die Energieeffizienz einer Architektur zu verbessern. Darüber hinaus sind sich die Teilnehmenden bewusst, dass man eine Reduzierung des CO<sub>2</sub>-Bedarfs durch die Verwendung von mehr erneuerbarer Energie oder durch eine Verringerung des Ressourcenbedarfs erzielen kann. Sie wissen, dass durch die Anwendung von Time- und Location Shifting der Anteil von grüner Energie erhöht werden kann, während andere Muster, wie beispielsweise Peak Shaving, den Ressourcenbedarf verringern.

#### LZ 6-5: Web Sustainability Guidelines kennen und einsetzen

Die Teilnehmenden kennen die Vorschläge des W3C Energieeffizienz, die beim Web Design zu berücksichtigen sind, wie beispielsweise Browser Caching, Einsatz eines CDNs, Edge Caching und sind in der Lage diese in ihren Anwendungen umzusetzen. Sie kennen unterschiedliche Dateiformate und Arten von graphischen Elementen (z. B. Animationen) und sind in der Lage deren Energieeffizienz einzuschätzen.

### 6.3. Referenzen

[GSF], [W3C]

## 7. Betrieb

Dauer: 75 Min.	Übungszeit: 30 Min.
----------------	---------------------

### 7.1. Begriffe und Konzepte

Power Usage Effectivness (PUE), Server Idle Energy Coefficient (SIEC), Service-Modelle, Deployment-Modelle, Cloud-Provider, Cloud-Migration

### 7.2. Lernziele

#### LZ 7-1: Energieeffizienz von Rechenzentren und Hardware beziffern

Die Teilnehmenden können die Energieeffizienz eines Rechenzentrums einschätzen. Sie kennen den Power Usage Effectivness (PUE) und seine Vor- und Nachteile.

Die Teilnehmenden kennen die Leistung-Last-Beziehung von Hardware. Sie können den Server Idle Energy Coefficient (SIEC) berechnen und einschätzen.

#### LZ 7-2: Cloud Service Modelle kennen und bezüglich Energieeffizienz beurteilen

Die Teilnehmenden kennen die Hauptkategorien von Cloud Computing Services (Cloud Service Modelle), insbesondere "Infrastructure as a Service", "Platform as a Service", "Software as a Service" und "Serverless". Sie können die Hauptcharakteristiken dieser Modelle benennen und abwägen wo und wie sie sich bzgl. Energieeffizienz und CO<sub>2</sub>-Emissionseffizienz unterscheiden und wo sie ähnlich einzuordnen sind.

#### LZ 7-3: Cloud Deployment Modelle kennen und bezüglich Energieeffizienz beurteilen

Die Teilnehmenden kennen die verschiedenen Deployment Modelle für Cloud Umgebungen wie vor Allem "Public Cloud", "Private Cloud", "Hybrid Cloud" und klassischer On-Premise-Betrieb. Sie können aufzeigen welche Chancen und Risiken diese Varianten auf Energieeffizienz und CO<sub>2</sub>-Emissionseffizienz haben. Dabei sind insbesondere die Energieeffizienz der Rechenzentren, die Flexibilität bei der Auswahl der Hardware, Überprovisionierung und Datenverkehr zu berücksichtigen.

#### LZ 7-4: Anbieter nach ökologisch-nachhaltigen Aspekten bewerten und auswählen

Die Teilnehmenden sind in der Lage ökologisch-nachhaltige Aspekte der verschiedenen Anbieter zu beurteilen, um diese in einer Auswahl berücksichtigen zu können. Dazu wissen sie, wie sie u. a. an folgende Informationen gelangen

- Informationen zum aktuellen Stand der Emissionen eines Anbieters (z. B. Sustainability Reports, Environmental Reports).
- Maßnahmen die Anbieter aktuell durchführen, um Emissionen zu reduzieren oder zu kompensieren.
- Roadmap und Klimaziele der Anbieter für die Zukunft.

Außerdem kennen die Teilnehmenden die Möglichkeiten und Einschränkungen der Anbieter beim Monitoring der Emissionen aus der eigenen Ressourcennutzung wie sie durch die Werkzeuge der Anbieter bereitgestellt werden.

#### LZ 7-5: Betriebliche Antipatterns kennen

Die Teilnehmenden kennen die Herausforderungen des energieeffizienten Betriebs und typische Anti-Pattern wie z. B Überprovisionierung durch fehlendes Monitoring, fehlende Automatisierung in der



Provisionierung oder ungünstige geografische Verteilung.

### **LZ 7-6: CO<sub>2</sub>-Optimierung in der Cloud vornehmen**

Die Teilnehmenden wissen, wie und in welchem Rahmen sie bei einzelnen Cloud-Anbietern die CO<sub>2</sub>-Emissionen beeinflussen können durch u. a.

- Auswahl von Zonen mit emissionsarmer Energieversorgung.
- Ressourcen sparsam dimensionieren und erst bei Bedarf skalieren.
- Bedarf zeitlich bzgl. Emissionen optimieren ("Time Shifting", "Peak Shaving").
- Autoscaling.
- Nutzung effizienter Datenspeicherung (nach Zugriffsweg und -häufigkeit).
- Reduktion bzw. Verkürzung des Netzwerkverkehrs.
- Serverless Computing.
- Nutzung von Managed Services.

### **7.3. Referenzen**

[PUE ], [energy efficiency report], [Ackermann et al. 2022]

## 8. Energieeffizienter Entwicklungsprozess

Dauer: 60 Min.	Übungszeit: keine
----------------	-------------------

### 8.1. Begriffe und Konzepte

Continuous Deployment, Continuous Integration, Deployment Pipelines, Testautomatisierung

### 8.2. Lernziele

#### LZ 8-1: CI/CD-Strategien und deren Ressourcenbedarf kennen

Die Teilnehmenden kennen verschiedene Strategien des Infrastructure as Code, Continuous Integration und Continuous Deployments. Dazu gehört, warum und wann diese üblicherweise eingesetzt werden, welchen grundsätzlichen Ressourcenbedarf diese haben können und wie sich Testausführung sowie Codeanalysen auf diesen Bedarf auswirken können.

#### LZ 8-2: Optimierung der CI/CD Prozesse durchführen

Die Teilnehmenden sind mit Methoden zur Ressourcenoptimierung von CI/CD-Prozessen vertraut. Beispiele hierfür sind die Anwendung von Peak-Shaving und Time-Shifting bei geplanten Builds, sowie das Deployment in räumlicher Nähe zur Zielgruppe im Kontext der Cloud.

#### LZ 8-3: Optimierung der Infrastruktur vornehmen

Die Teilnehmenden kennen Verfahren zur Optimierung der Entwicklungs- und Deployment-Infrastruktur. Dazu zählen beispielsweise Betrachtung der tatsächlich notwendigen Resilienz, die Effizienzvorteile von Containern gegenüber virtuellen Maschinen, sowie der Einsatz von Caching für Build- und Testartefakte.

#### LZ 8-4: Teststrategie optimieren

Die Teilnehmenden sind mit Möglichkeiten zur Verbesserung der Ressourceneffizienz ihrer Teststrategien und Testumgebungen vertraut. Dazu zählt beispielsweise die Wirkung verschiedener Testarten (Last-Test, Systemtest, Integrationstest, Unit-Test, ...) auf den Ressourcenverbrauch sowie die Verringerung dessen durch Test-Gap-Analysen und die bedarfsorientierte, zeitlich begrenzte Bereitstellung entsprechender Testumgebungen.

## Referenzen

Dieser Abschnitt enthält Quellenangaben, die ganz oder teilweise im Curriculum referenziert werden.

### A

- [arc42] arc42 Quality Model, <https://quality.arc42.org>
- [PUE ] Avelar et al.: A Comprehensive Examination of the Metric, [https://datacenters.lbl.gov/sites/default/files/WP49-PUE%20A%20Comprehensive%20Examination%20of%20the%20Metric\\_v6.pdf](https://datacenters.lbl.gov/sites/default/files/WP49-PUE%20A%20Comprehensive%20Examination%20of%20the%20Metric_v6.pdf)

### B

- [bitkom 2021] bitkom: Klimaeffekte der Digitalisierung, [https://www.bitkom.org/sites/main/files/2021-10/20211010\\_bitkom\\_studie\\_klimaeffekte\\_der\\_digitalisierung.pdf](https://www.bitkom.org/sites/main/files/2021-10/20211010_bitkom_studie_klimaeffekte_der_digitalisierung.pdf)
- [Brundtland 1987] Brundtland, G. (1987): Report of the World Commission on Environment and Development: Our Common Future, <http://www.un-documents.net/wced-ocf.htm>

### C

- [Clements et al. 2003] Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers et al.: Documenting Software Architectures – Views and Beyond. Addison Wesley, 2003.
- [Cloud Carbon Footprint] Cloud Carbon Emissions Measurement and Analysis Tool, <https://www.cloudcarbonfootprint.org>
- [Code Carbon] Python based tool to estimate the amount of carbon dioxide (CO2) , <https://codecarbon.io>

### E

- [energy efficiency report] Energy-efficient Cloud Computing Technologies and Policies for an Eco-friendly Cloud Market, <https://digital-strategy.ec.europa.eu/en/library/energy-efficient-cloud-computing-technologies-and-policies-eco-friendly-cloud-market>

### G

- [Greenframe] Measure and reduce website's CO2 emissions, <https://greenframe.io>
- [Green Metrics Tool] Messen des Energie- / CO2-Verbrauch von Software-Architekturen, <https://www.green-coding.io/de/projects/green-metrics-tool/>
- [GSF] Green Software Design Patterns <https://patterns.greensoftware.foundation/>
- [Greenhouse Gas Protocol] Greenhouse Gas Protocol, <https://learn.greensoftware.foundation/measurement>

### J

- [JoularJX] Java-based agent for power monitoring at the source code level, <https://www.noureddine.org/research/joular/joularjx>

### P

- [Pereira et al.] Rui Pereira, Marco Couto, Francisco Ribeiro, Rui Rua, Jácome Cunha, João Paulo

Fernandes, João Saraiva (2021): Ranking Programming Languages by Energy Efficiency

## Q

- [ISO 25010] ISO/IEC 25010, <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

## S

- [SCI] Software Carbon Intensity, <https://sci-guide.greensoftware.foundation/>
- [Ackermann et al. 2022] Dr. Ludger Ackermann und Dr. Dirk Harryvan: Auszeit unter Aufsicht in iX Special 2022 - Green IT, <https://shop.heise.de/ix-13-2022/PDF>

## W

- [W3C] Web Sustainability Guidelines (WSG) 1.0, <https://w3c.github.io/sustyweb/>