

Curriculum for

Certified Professional for
Software Architecture (CPSA)[®]
Advanced Level

**Module
GREEN**

**Green Software - Development of resource-efficient
applications**

2024.1-rev1-EN-20250506



Table of Contents

| | |
|--|----|
| List of Learning Goals | 2 |
| Introduction: General information about the iSAQB Advanced Level | 3 |
| What is taught in an Advanced Level module? | 3 |
| What can Advanced Level (CPSA-A) graduates do? | 3 |
| Requirements for CPSA-A certification | 3 |
| Essentials | 4 |
| What does the module “GREEN” convey? | 4 |
| Curriculum Structure and Recommended Durations | 4 |
| Duration, Teaching Method and Further Details | 4 |
| Prerequisites | 5 |
| Structure of the Curriculum | 5 |
| Supplementary Information, Terms, Translations | 5 |
| 1. Climate change, opportunities of digitalisation, role of IT, basic concepts | 6 |
| 1.1. Terms and Principles | 6 |
| 1.2. Learning Goals | 6 |
| 1.3. References | 6 |
| 2. Principles, regulation and effects on companies | 7 |
| 2.1. Terms and Principles | 7 |
| 2.2. Learning Goals | 7 |
| 2.3. References | 7 |
| 3. Quality | 8 |
| 3.1. Terms and Principles | 8 |
| 3.2. Learning Goals | 8 |
| 3.3. References | 8 |
| 4. Measuring and Monitoring | 9 |
| 4.1. Terms and Principles | 9 |
| 4.2. Learning Goals | 9 |
| 4.3. References | 10 |
| 5. Software development | 11 |
| 5.1. Terms and Principles | 11 |
| 5.2. Learning Goals | 11 |
| 5.3. References | 11 |
| 6. Software architecture | 12 |
| 6.1. Terms and Principles | 12 |
| 6.2. Learning Goals | 12 |
| 6.3. References | 12 |
| 7. Operations | 13 |



7.1. Terms and Principles 13

7.2. Learning Goals 13

7.3. References 14

8. Energy-efficient development process 15

8.1. Terms and Principles 15

8.2. Learning Goals 15

References 16

© (Copyright), International Software Architecture Qualification Board e. V. (iSAQB® e. V.) 2025

The curriculum may only be used subject to the following conditions:

1. You wish to obtain the CPSA Certified Professional for Software Architecture Foundation Level® certificate or the CPSA Certified Professional for Software Architecture Advanced Level® certificate. For the purpose of obtaining the certificate, it shall be permitted to use these text documents and/or curricula by creating working copies for your own computer. If any other use of documents and/or curricula is intended, for instance for their dissemination to third parties, for advertising etc., please write to info@isaqb.org to enquire whether this is permitted. A separate license agreement would then have to be entered into.
2. If you are a trainer or training provider, it shall be possible for you to use the documents and/or curricula once you have obtained a usage license. Please address any enquiries to info@isaqb.org. License agreements with comprehensive provisions for all aspects exist.
3. If you fall neither into category 1 nor category 2, but would like to use these documents and/or curricula nonetheless, please also contact the iSAQB e. V. by writing to info@isaqb.org. You will then be informed about the possibility of acquiring relevant licenses through existing license agreements, allowing you to obtain your desired usage authorizations.

Important Notice

We stress that, as a matter of principle, this curriculum is protected by copyright. The International Software Architecture Qualification Board e. V. (iSAQB® e. V.) has exclusive entitlement to these copyrights.

The abbreviation "e. V." is part of the iSAQB's official name and stands for "eingetragener Verein" (registered association), which describes its status as a legal entity according to German law. For the purpose of simplicity, iSAQB e. V. shall hereafter be referred to as iSAQB without the use of said abbreviation.

List of Learning Goals

- LG 1-1: Recognising and addressing the global challenges of climate change
- LG 1-2: Knowing ways to save energy through digitalisation
- LG 1-3: Categorising and quantifying IT energy consumption and know its drivers
- LG 1-4: Knowing the fields of action for saving CO₂
- LG 2-1: Knowing the regulations and their requirements
- LG 2-2: Knowing and applying the Greenhouse Gas Protocol
- LG 2-3: Positioning energy efficiency as a field of action in the company
- LG 2-4: Knowing the stakeholders in the context of green IT and their prioritisation
- LG 3-1: Knowing quality models and their relation to energy efficiency
- LG 3-2: Dealing with the interactions between quality attributes and energy efficiency
- LG 4-1: Bringing resources in relation to the service provided
- LG 4-2: Knowing and using measuring methods and tools
- LG 4-3: Knowing and applying measurement methods
- LG 4-4: Knowing integration options for measuring energy efficiency in the development process
- LG 5-1: Know ways to increase energy efficiency in software development
- LG 5-2: Know and apply procedures for energy-efficient data handling
- LG 5-3: Know the effects and possibilities of optimisations
- LG 6-1: Knowing and understanding architectural styles and their relation to energy efficiency
- LG 6-2: Knowing communication and its impact on energy efficiency
- LG 6-3: Knowing databases and their characteristics in relation to energy efficiency
- LG 6-4: Knowing and applying specific Green IT patterns
- LG 6-5: Knowing and using Web Sustainability Guidelines
- LG 7-1: Quantifying the energy efficiency of data centres and hardware
- LG 7-2: Knowing cloud service models and assessing them in terms of energy efficiency
- LG 7-3: Knowing cloud deployment models and assessing them in terms of energy efficiency
- LG 7-4: Evaluating and selecting suppliers according to ecologically sustainable aspects
- LG 7-5: Knowing operational antipatterns
- LG 7-6: Carrying out CO₂ optimisation in the cloud
- LG 8-1: Knowing CI/CD strategies and their resource requirements
- LG 8-2: Optimizing CI/CD processes
- LG 8-3: Optimising the infrastructure
- LG 8-4: Optimising the test strategy

Introduction: General information about the iSAQB Advanced Level

What is taught in an Advanced Level module?

- The iSAQB Advanced Level offers modular training in three areas of competence with flexibly designable training paths. It takes individual inclinations and priorities into account.
- The certification is done as an assignment. The assessment and oral exam is conducted by experts appointed by the iSAQB.

What can Advanced Level (CPSA-A) graduates do?

CPSA-A graduates can:

- Independently and methodically design medium to large IT systems
- In IT systems of medium to high criticality, assume technical and content-related responsibility
- Conceptualize, design, and document actions to achieve quality requirements and support development teams in the implementation of these actions
- Control and execute architecture-relevant communication in medium to large development teams

Requirements for CPSA-A certification

- Successful training and certification as a Certified Professional for Software Architecture, Foundation Level® (CPSA-F)
- At least three years of full-time professional experience in the IT sector; collaboration on the design and development of at least two different IT systems
 - Exceptions are allowed on application (e.g., collaboration on open source projects)
- Training and further education within the scope of iSAQB Advanced Level training courses with a minimum of 70 credit points from at least three different areas of competence
- Successful completion of the CPSA-A certification exam



Essentials

What does the module “GREEN” convey?

While IT was long seen as the solution to many problems associated with climate change, it has now itself become the subject of optimisation considerations. Inefficient programming, often caused by the need for a fast time-to-market, has often been compensated for by ever faster hardware or more resources in the cloud. This path must be abandoned. The CO₂ emissions caused by software must be consistently reduced.

In the GREEN module, participants learn to take a holistic view of the topic of green software. This begins with an examination of the role of IT in halting climate change, an introduction to current regulations, a view of the requirements of various stakeholders and the identification of fields of action in companies. From there, we move on to measuring and monitoring CO₂ emissions and energy consumption and on to the core topic of software development. This central topic area includes the energy efficiency of various software architectures and basic concepts used, energy-efficient data handling, optimised algorithms and also the influence and management of quality requirements in relation to energy efficiency. Another important component is the topic of the cloud, both in terms of provider selection and the possibilities for low-CO₂ operation. Finally, options for improving energy efficiency in the development process are considered.

At the end of the module GREEN, participants will know the key levers for reducing CO₂ in IT. They will be able to assess the impact of software architecture and how architecture patterns and styles work. They know the pitfalls of data handling and can assess the impact of the selection of technical components. With regard to the cloud, participants know how to utilise the options offered there in terms of energy efficiency.

Curriculum Structure and Recommended Durations

| Content | Recommended minimum duration (minutes) |
|--|--|
| 1. Climate change, opportunities of digitalisation, role of IT, basic concepts | 60 |
| 2. Principles, regulation and effects on companies | 60 |
| 3. Quality | 60 |
| 4. Measurement and monitoring | 150 |
| 5. Software development | 75 |
| 6. Software architecture | 150 |
| 7. Operation | 105 |
| 8. Energy-efficient development process | 60 |
| Total | 720 (12h) |

Duration, Teaching Method and Further Details

The times stated below are recommendations. The duration of a training course on the GREEN module should be at least 2 days, but may be longer. Providers may differ in terms of duration, teaching method, type and structure of the exercises, and the detailed course structure. In particular, the curriculum provides no specifications on the nature of the examples and exercises.

Licensed training courses for the GREEN module contribute the following credit points towards admission to the final Advanced Level certification exam:

| | |
|---------------------------|-----------|
| Methodical Competence: | 10 Points |
| Technical Competence: | 10 Points |
| Communicative Competence: | 0 Points |

Prerequisites

Participants **should** have the following prerequisite knowledge:

- Practical experience in the design and development of small to medium-sized software systems
- Knowledge of the life cycle of software systems
- Dealing with quality requirements
- Practical experience in monitoring software systems
- Practical programming experience

Knowledge in the following areas may be **helpful** for understanding some concepts:

- First practical experience with performance engineering
- First practical experience with common cloud providers
- First practical experience with CI/CD pipelines

Structure of the Curriculum

The individual sections of the curriculum are described according to the following structure:

- **Terms/principles:** Essential core terms of this topic.
- **Teaching/practice time:** Defines the minimum amount of teaching and practice time that must be spent on this topic or its practice in an accredited training course.
- **Learning goals:** Describes the content to be conveyed including its core terms and principles.

This section therefore also outlines the skills to be acquired in corresponding training courses.

Supplementary Information, Terms, Translations

To the extent necessary for understanding the curriculum, we have added definitions of technical terms to the [iSAQB glossary](#) and complemented them by references to (translated) literature.

1. Climate change, opportunities of digitalisation, role of IT, basic concepts

| | |
|------------------|---------------------|
| Duration: 60 min | Practice time: none |
|------------------|---------------------|

1.1. Terms and Principles

Climate change, consumption of IT, savings through IT, savings in IT, sustainability, CO₂, CO₂-equivalents, CO₂ footprint, CO₂ handprint, negative external effects, watts, joules, efficiency, effectiveness

1.2. Learning Goals

LG 1-1: Recognising and addressing the global challenges of climate change

Participants are familiar with the global challenges of climate change. They know the increasing energy and resource requirements of IT and their main drivers. They can explain why IT must also address the issue of resource and energy efficiency.

LG 1-2: Knowing ways to save energy through digitalisation

Participants know the possibilities for saving energy and CO₂ through digitalisation. They know the sectors that particularly benefit from digitalisation. The difference between savings in IT and through IT is known.

LG 1-3: Categorising and quantifying IT energy consumption and know its drivers

Participants can quantify the energy consumption of information technology, categorise it in relation to other sectors and name the main drivers. They are familiar with trends and developments in IT and key levers.

LG 1-4: Knowing the fields of action for saving CO₂

The participants know the fields of action for saving CO₂:

- Energy efficiency - Consume as little energy as possible.
- Hardware efficiency - Use as little hardware as possible.
- CO₂ emission efficiency - Consume energy when and where it is generated in the "greenest" way.

1.3. References

[bitkom 2021], [Brundtland 1987]

2. Principles, regulation and effects on companies

| | |
|------------------|---------------------|
| Duration: 60 min | Practice time: none |
|------------------|---------------------|

2.1. Terms and Principles

Energy efficiency, certifications (e.g. Blue Angel, TCO), regulatory requirements (e.g. CSRD, ESD), Greenhouse Gas Protocol (GHG Protocol)

2.2. Learning Goals

LG 2-1: Knowing the regulations and their requirements

Participants know the regulatory framework and the requirements it contains for saving CO₂. They know the existing certifications, such as the Blue Angel, and can use them appropriately in their work. Participants are familiar with standards such as ESG (Environmental Social Governance) and CSRD (Corporate Sustainability Reporting Directive) and their effects and consequences for companies.

LG 2-2: Knowing and applying the Greenhouse Gas Protocol

Participants are familiar with the Greenhouse Gas Protocol (GHP). They are able to explain the scopes and assign emissions to the scopes. They can explain which scopes are affected when operating software in the cloud or on premise. Participants are familiar with ISO 14064-1 as a derivation of the Greenhouse Gas Protocol.

LG 2-3: Positioning energy efficiency as a field of action in the company

The participants know the motivation for placing green software in the company and the arguments and addressees for successfully anchoring it in the company. They can set sensible targets for the reduction of greenhouse gases for companies. Participants can explain what requirements third parties have along the value chain and how these affect their own company.

LG 2-4: Knowing the stakeholders in the context of green IT and their prioritisation

Participants know the requirements of the various stakeholders for energy-efficient software and the resulting fields of action. They know the areas with the greatest leverage for reducing greenhouse gases and can prioritize measures accordingly. Participants can classify different software systems and recognize the costs and benefits of optimization.

2.3. References

[\[Greenhouse Gas Protocol\]](#)

3. Quality

| | |
|------------------|-----------------------|
| Duration: 45 min | Practice time: 15 min |
|------------------|-----------------------|

3.1. Terms and Principles

Quality model, quality scenarios, quality objectives, ISO 25010

3.2. Learning Goals

LG 3-1: Knowing quality models and their relation to energy efficiency

Participants are familiar with different quality models and can describe quality attributes relating to energy efficiency and categorise them in the quality models. In addition, participants are able to formulate quality scenarios for quality attributes in the area of energy efficiency. Depending on the intended use and the quantity structure of the application, they can consciously decide whether the quality criteria are used only for the application or for the optimisation of the development process.

LG 3-2: Dealing with the interactions between quality attributes and energy efficiency

Participants know the positive and negative interactions between quality attributes in the area of energy efficiency and other quality attributes. Based on this knowledge, they can recognise quality requirements that conflict with each other and develop alternative solutions to mitigate conflicts. In particular, the quality attributes of elasticity and scalability, modularity and resilience are discussed in more detail so that participants can assess their influence on energy efficiency.

3.3. References

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

4. Measuring and Monitoring

| | |
|------------------|-----------------------|
| Duration: 90 Min | Practice time: 60 min |
|------------------|-----------------------|

4.1. Terms and Principles

Software Carbon Intensity (SCI), business metrics, proxy metrics, energy efficiency, categories of measurement methods, measurement tools

4.2. Learning Goals

LG 4-1: Bringing resources in relation to the service provided

Participants are able to relate resources to the service provided (business metric). They are familiar with the ISO standard Software Carbon Intensity (SCI) as a metric and understand how it can be used as a business metric.

LG 4-2: Knowing and using measuring methods and tools

Participants know different categories of measurement methods and their areas of application. They understand how to combine tools from the different categories for an analysis. Categories include, for example:

- Hardware measurement tools. Measure consumption directly on the hardware.
- Full-stack measurement tools. Measure all components of an application such as consumption per Docker container, network.
- Programming language-specific measurement tools. Provide "whitebox" measurements within an application.
- Measurement tools for web applications. Provide information about the network load generated by a page request, among other things.
- Carbon footprint tools from cloud providers. In cloud environments, energy consumption cannot be measured directly. Instead, you have to rely on tools provided by the provider or other estimation methods. Participants are familiar with the functionality and limitations of these tools (inaccuracy, relative data, necessary authorisations in the cloud console) as well as the challenges of comparing managed services with pure VMs.

LG 4-3: Knowing and applying measurement methods

The participants know the methodological procedure for measuring energy consumption and CO₂ intensity. They understand that it is generally not possible to evaluate/measure absolute consumption very precisely (especially in the cloud), but how to analyse trends instead. They know proxy metrics for CO₂ intensity.

Participants know how to set up meaningful scenarios for measuring energy efficiency.

- They know criteria for designing the scenarios (e.g. common processes, prioritisation according to frequency of use).
- They know which interfaces to use depending on the objective of the measurement (end-to-end scenarios, using APIs, etc.).
- They know tools for creating such scenarios.

- They can deal adequately with measurement inaccuracies (e.g. running scenarios multiple times, interpreting deviations).

Participants understand that the procedure for measuring energy efficiency is related to the procedure for measuring performance. Participants understand how to proceed systematically in order to narrow down consumption hotspots.

LG 4-4: Knowing integration options for measuring energy efficiency in the development process

Participants know how to integrate energy efficiency measurements into the development process. They know how to integrate the measurements into the CI/CD pipeline. They understand how to find a good balance for the frequency of the measurements - on the one hand (negative) trends should be recognised quickly, on the other hand the measurement itself also consumes energy.

Participants understand that the measurements must be as simple and lightweight as possible for developers to use in order to increase acceptance. They understand that energy efficiency measurements compete with quick feedback from the build. You understand that the following points are also important for acceptance:

- Creating the scenarios for the measurement must be easy. It is advantageous to use tools that are already established in the project (e.g. frameworks for creating end-to-end tests)
- The presentation of the measurement results must be clear, easy and quick to understand. In particular, the most important results such as trends or significant deviations in consumption must be directly recognisable.

4.3. References

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

5. Software development

| | |
|------------------|---------------------|
| Duration: 75 min | Practice time: none |
|------------------|---------------------|

5.1. Terms and Principles

Programming languages, ahead-of-time and just-in-time compilation, bytecode, machine code, garbage collection, data structures, algorithms, computational complexity, caching, database systems, data models

5.2. Learning Goals

LG 5-1: Know ways to increase energy efficiency in software development

Participants know the procedures and limits for optimising programs. They can distinguish sensible starting points from less worthwhile goals. Participants are familiar with the energy efficiency of various programming languages and can assess the areas of application for which they are suitable. They can evaluate the runtime behaviour and development speed of programming languages and make the right choice based on the requirements. They also understand the differences between ahead-of-time and just-in-time compilation, bytecode and machine code and the conflicting goals of garbage collectors.

LG 5-2: Know and apply procedures for energy-efficient data handling

Participants know methods for handling large amounts of data. They can assess data structures and operations on these data structures in terms of energy efficiency and make the right selection based on the given requirements. Participants know the pitfalls of using database systems and can avoid them. They are able to evaluate and optimally select different approaches for mapping relationships between entities on the basis of business requirements and quality requirements.

LG 5-3: Know the effects and possibilities of optimisations

Participants are familiar with the principle of computational complexity and can evaluate and select algorithms in terms of their runtime and energy efficiency. They can use caches in connection with energy efficiency, know how they work (e.g. cache replacement) as well as the advantages and disadvantages. They know the possibilities and limitations of tools for detecting energy code smells.

5.3. References

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

6. Software architecture

| | |
|------------------|-----------------------|
| Duration: 90 min | Practice time: 60 min |
|------------------|-----------------------|

6.1. Terms and Principles

Architecture styles, monolith, microservices, serverless, inter-process communication, databases, green IT patterns

6.2. Learning Goals

LG 6-1: Knowing and understanding architectural styles and their relation to energy efficiency

Participants can assess different architectural styles such as distributed systems, structured monoliths or serverless and their impact on energy efficiency. In addition, they are able to assess the difference between stateless and stateful components in terms of energy requirements. Participants are familiar with the principles of cloud-native architectures and their impact on energy efficiency.

LG 6-2: Knowing communication and its impact on energy efficiency

Participants are familiar with different types of communication (synchronous versus asynchronous) between building blocks in relation to the provision of data and the formats and protocols used (text-based versus binary protocols) with regard to energy efficiency. They are able to evaluate the influence of data compression on the transferred data volume and know that energy efficiency is positively influenced by a reduction in call frequency and data volume.

LG 6-3: Knowing databases and their characteristics in relation to energy efficiency

Participants are familiar with different types of database models (relational, NoSQL, etc.) and different database systems (Postgres, DB2, Oracle, etc.) and can make the right choice in terms of energy efficiency. They are able to select a suitable service model (self-hosting versus cloud-based managed service) for their data storage.

LG 6-4: Knowing and applying specific Green IT patterns

Participants are generally familiar with patterns such as those of the Green Software Foundation to improve the energy efficiency of an architecture. In addition, the participants are aware that a reduction in CO₂ requirements can be achieved by using more renewable energy or by reducing resource requirements. They know that by applying time and location shifting, the proportion of green energy can be increased while other patterns, such as peak shaving, reduce resource requirements.

LG 6-5: Knowing and using Web Sustainability Guidelines

Participants are familiar with the W3C energy efficiency proposals to be considered in web design, such as browser caching, use of a CDN, edge caching and are able to implement these in their applications. They are familiar with different file formats and types of graphic elements (e.g. animations) and are able to assess their energy efficiency.

6.3. References

[GSF], [W3C]

7. Operations

| | |
|------------------|-----------------------|
| Duration: 75 min | Practice time: 30 min |
|------------------|-----------------------|

7.1. Terms and Principles

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

7.2. Learning Goals

LG 7-1: Quantifying the energy efficiency of data centres and hardware

Participants can assess the energy efficiency of a data centre. They know the Power Usage Effectiveness (PUE) and its advantages and disadvantages.

Participants know the power-load relationship of hardware. They can calculate and estimate the Server Idle Energy Coefficient (SIEC).

LG 7-2: Knowing cloud service models and assessing them in terms of energy efficiency

Participants know the main categories of cloud computing services (cloud service models), in particular "Infrastructure as a Service", "Platform as a Service", "Software as a Service" and "Serverless". They can name the main characteristics of these models and weigh up where and how they differ in terms of energy efficiency and CO₂ emission efficiency and where they are similar.

LG 7-3: Knowing cloud deployment models and assessing them in terms of energy efficiency

Participants know the different deployment models for cloud environments, in particular "public cloud", "private cloud", "hybrid cloud" and classic on-premise operation. They can demonstrate the opportunities and risks of these variants in terms of energy efficiency and CO₂ emission efficiency. In particular, the energy efficiency of data centres, flexibility in the selection of hardware, overprovisioning and data traffic must be taken into account.

LG 7-4: Evaluating and selecting suppliers according to ecologically sustainable aspects

Participants are able to assess ecologically sustainable aspects of the various providers in order to be able to take these into account when making a selection. To this end, they know how to obtain the following information, among others

- Information on the current status of a provider's emissions (e.g. sustainability reports, environmental reports).
- Measures that providers are currently implementing to reduce or offset emissions.
- Roadmap and climate targets of providers for the future.

In addition, the participants know the possibilities and limitations of the providers in monitoring emissions from their own resource use as provided by the providers' tools.

LG 7-5: Knowing operational antipatterns

Participants are familiar with the challenges of energy-efficient operation and typical anti-patterns such as overprovisioning due to a lack of monitoring, a lack of automation in provisioning or unfavourable geographical distribution.

LG 7-6: Carrying out CO₂ optimisation in the cloud

Participants know how and to what extent they can influence the CO₂ emissions of individual cloud providers by, among other things

- Selecting zones with low-emission energy supply.
- Dimensioning resources sparingly and only scaling them when required.
- Optimising demand in terms of emissions ("time shifting", "peak shaving").
- Autoscaling.
- Utilise efficient data storage (according to access path and frequency).
- Reduction or shortening of network traffic.
- Serverless computing.
- Use of managed services.

7.3. References

[\[PUE \]](#), [\[energy efficiency report\]](#), [\[Ackermann et al. 2022\]](#)

8. Energy-efficient development process

| | |
|------------------|---------------------|
| Duration: 60 min | Practice time: none |
|------------------|---------------------|

8.1. Terms and Principles

Continuous Deployment, Continuous Integration, Deployment Pipelines, Test automation

8.2. Learning Goals

LG 8-1: Knowing CI/CD strategies and their resource requirements

Participants are familiar with various strategies of Infrastructure as Code, Continuous Integration and Continuous Deployments. This includes why and when these are usually used, what basic resource requirements they can have and how test execution and code analyses can affect these requirements.

LG 8-2: Optimizing CI/CD processes

Participants are familiar with methods for resource optimisation of CI/CD processes. Examples of this include the use of peak shaving and time shifting for planned builds, as well as deployment in close proximity to the target group in the context of the cloud.

LG 8-3: Optimising the infrastructure

Participants know methods for optimising the development and deployment infrastructure. This includes, for example, consideration of the actual resilience required, the efficiency advantages of containers over virtual machines and the use of caching for build and test artefacts.

LG 8-4: Optimising the test strategy

Participants are familiar with ways to improve the resource efficiency of their test strategies and test environments. This includes, for example, the effect of different test types (load test, system test, integration test, unit test, ...) on resource consumption and the reduction of this through test gap analyses and the demand-oriented, time-limited provision of appropriate test environments.

References

This section contains references that are cited in the curriculum.

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

B

- [bitkom 2021] bitkom: Klimaeffekte der Digitalisierung, 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/>