Sustainable Testing

White Paper

Alessandro Sebaste (STB)
Armin Beer (ATB)
Dietrich Leimsner (GTB)
Lilia Gargouri (GTB)
Reto Armuzzi (STB)
Stefan Mohacsi (ATB)

Contents

Sustainable Testing 1
1 Introduction 3
1.1 Management Summary 3
1.2 Energy Consumption of the ICT Sector 3
1.3 What is Sustainability? 5
1.4 Motivation 6
2 Analysis of Relevant ISTQB Syllabi 7
2.1 Goals 7
2.2 Approach 7
2.3 Overall Results 8
2.4 Results for CTFL 11
3 Suggestions for Augmenting Sustainability Aspects in ISTQB Syllabi 12
3.1 Certified Tester Foundation Level Syllabus 12
3.2 Other Relevant Syllabi 13
4 UN Sustainable Development Goals 14
4.1 Overview 14
4.2 Impact of Testing 15
5 Sustainability vs. AI 17
5.1 Conflicts 17
5.2 Conclusion 18
6 Final Thoughts and Future Work 19
7 References 19

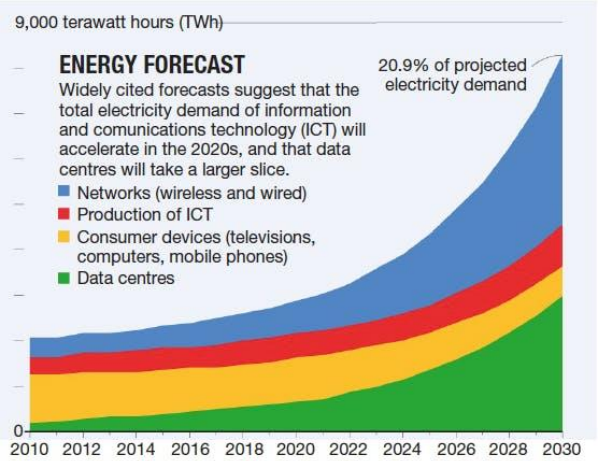
1 Introduction

1.1 Management Summary

In this white paper, we want to raise awareness for the different aspects of sustainability and their relationship to software testing. We analyse to what extent these aspects are already covered in the existing ISTQB syllabi and give a number of suggestions for improvements. We also concern ourselves with the impact of testing on the UN Sustainable Development Goals and potential conflicts that should be considered when deciding about the use of AI.

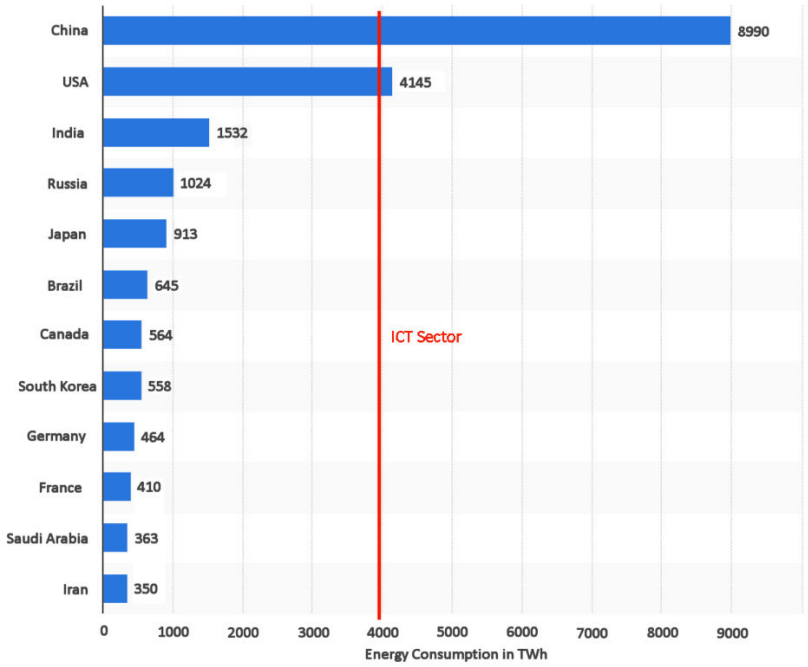
1.2 Energy Consumption of the ICT Sector

The combination of exponential data growth, energy-intensive AI workloads, a flood of IoT devices, limited net efficiency gains (rebound effect), and the ongoing expansion of communications infrastructure is driving up ICT energy consumption. If these trends continue unabated, the ICT sector could indeed become the largest single energy consumer worldwide by 2030.



Source: Power, Efficiency, and Sustainability Trends

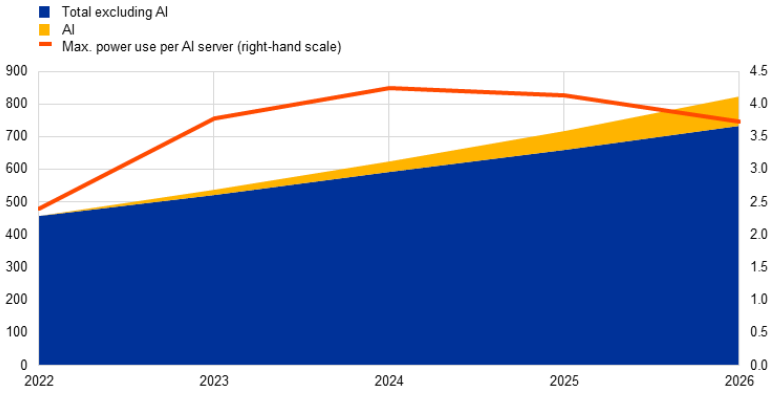
In 2024, only two countries consumed more electricity than the global ICT sector.



Source: Largest countries worldwide by electricity consumption in 2024

The required energy comes from an energy mix that is largely, around two-thirds, derived from non-renewable energy sources. As a result, the software in the ICT industry already accounts for a significant proportion of greenhouse gas emissions. Thus, there is a clear need to discuss the issue of sustainability in the ICT industry.

In addition, ICT systems are becoming increasingly complex and interconnected. This requires development and test, especially when it comes to ensuring the quality of services. However, it is impossible to test everything or to guarantee that something works as expected. We must view quality as a strategic goal.



Source: IEA, Goldman Sachs and ECB staff calculations
Projected growth in the electricity demand of data centres

1.3 What is Sustainability?

1.3.1 Definition

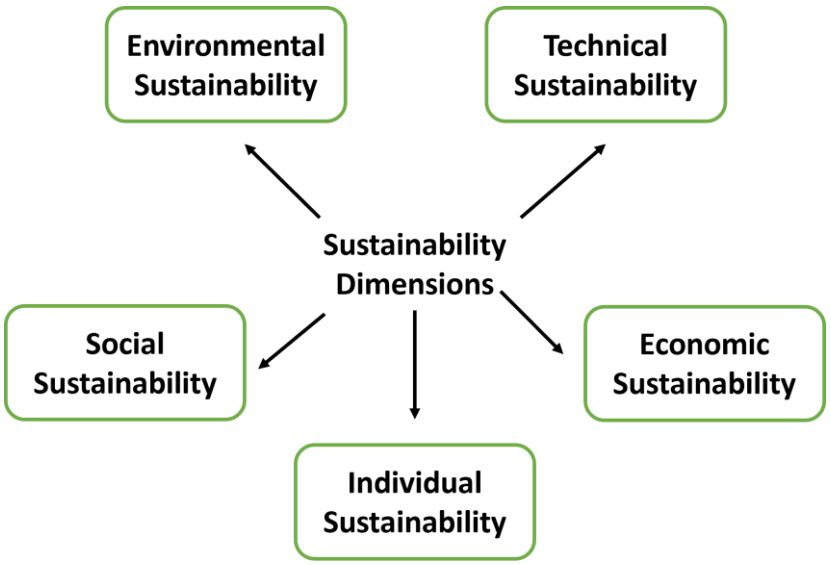
The term *sustainability* derives from the Latin word *sustinere* which means *protecting*, *preserving* and *retaining*. The Brundtland Commission [1] defines *sustainability* as follows:

"[..] meeting the needs of the present without compromising the ability of future generations to meet their own needs"

Typical goals include avoiding the waste of non-renewable resources, protecting the climate, and creating durable and maintainable products.

1.3.2 Sustainability Dimensions

Penzenstadler and Femmer [2] propose a generic sustainability framework for software engineering to improve sustainability in organizations. There are five distinct sustainability dimensions: environmental, technical, economic, individual, and social sustainability.



The framework's practical application consists of two distinct phases: analysis and application & assessment. The analysis phase consists of instantiating the generic sustainability model, and the application & assessment phase consists of specifying responsibilities and monitoring sustainability using previously defined indicators and metrics.

1. **Environmental sustainability:** This is probably the most well-known dimension. Protecting the environment and Earth's climate is also relevant to the testing process, where hardware resources are used that might consume vast amounts of energy. The "GREENSOFT Model" [3] provides a conceptual reference model and guidance that can be used as a starting point.
2. **Technical sustainability:** Maintainability and ease of use in future updates should be key considerations in technical decisions. A lack of testability leads to

an increase in complexity from version to version and reduces the comprehensibility and quality of test cases. For instance, a lack of knowledge sharing between developers and testers, and a lack of technology management lead to flawed decisions in long-term planning.

3. **Economic sustainability:** This should be assessed in advance of any IT project or financial decision. It can be achieved primarily by considering the anticipated costs and expected benefits, for example, when deciding about the introduction of test automation. This investment might cause considerable costs at the start but pays off after several test runs. What is more, NOT making this investment might cause enormous costs later and jeopardize the whole project.
4. **Individual Sustainability:** From the perspective of a project team member, work-life balance, career opportunities, and job satisfaction are more important than ever. When recruiting new employees, it's crucial to offer them opportunities to implement innovations and engage in interesting, long-term projects. Prioritizing a healthy work-life balance and employee well-being can foster long-term commitment to the company.
5. **Social Sustainability:** Social sustainability refers to the interaction of all stakeholders throughout the organization and within the test team. It encompasses the recruitment of suitable employees, the maturity of the testing process, flexible working arrangements, and team cohesion.

1.4 Motivation

What does sustainability have to do with software development and testing? A simple example is, protecting the environment by considering energy efficiency in software design and test planning. Such **sustainability engineering** requires a paradigm shift in several factors, including project organization, team structure, and prioritization of quality characteristics.

There are two main questions that need to be addressed:

- How can software testing be performed in a sustainable way?
  Since software testing requires resources, it is important to do it in a way that is efficient and effective. Therefore, sustainability should be considered during test planning as well as in later phases of the test process.
- How can the sustainability of a software system be tested?
  If we consider sustainability to be a quality characteristic, we can also test the software for this characteristic. By testing we can demonstrate and measure sustainability.

Sustainable Software Testing

This refers to practices and strategies used in software testing that aim at reducing resource consumption and environmental impact. It involves creating efficient test processes that minimize energy use, optimize hardware utilization, and reduce the carbon footprint of testing activities. The goal is to achieve software quality while being mindful of ecological sustainability.

Testing the Sustainability of Software Systems

This focuses on evaluating how environmentally friendly a piece of software is over its entire lifecycle. It includes assessing factors such as energy efficiency, resource consumption, and the software’s impact on hardware longevity. The testing process ensures that the software adheres to sustainability principles, such as reducing energy waste during execution and promoting efficient use of system resources.

In this white paper, we will mainly focus on the first point, sustainable software testing. It is structured as follows:

- In Chapter 2, we present an analysis of the current ISTQB Syllabi regarding their relevance to sustainability
- In Chapter 3, we give suggestions for possible augmentations of the sustainability aspects in the ISTQB syllabi
- Chapter 4 introduces the UN Sustainable Development Goals and their relationship to testing
- Chapter 5 shows how the ever-growing use of AI, both in software testing and in general, affects sustainability
- Chapter 6 gives a short summary and an outlook on future work

2 Analysis of Relevant ISTQB Syllabi

2.1 Goals

In order to answer the question how testing can be performed in a sustainable way, we decided to analyse the test activities and processes in the ISTQB syllabi regarding their relationship with sustainability.

More precisely, we wanted to find out the following:

- To which extent do the current ISTQB syllabi cover topics related to sustainable software testing?
- Which (sub)chapters of the ISTQB syllabi are especially relevant to sustainable software testing?
- Which of the sustainability dimensions is addressed in which (sub)chapter?
- Which phase of the test process is addressed in which (sub)chapter?
- Are there any relevant points regarding sustainable testing that are still missing?

2.2 Approach

We examined those syllabi of the ISTQB curriculum that seemed to have the highest probability of having a relevant impact on sustainability. For each of the selected syllabi, we mapped the applicable subchapters to test phases and sustainability dimensions. We focused on economic, technical, and environmental sustainability but left out the social and individual dimensions because they are covered in less detail in the syllabi.

Syllabus	Chapter Number	Chapter titel	Economic sustainability	Technical sustainability	Environmental sustainability	Test planning	Test monitoring And test control	Test analysis
CTFL	1.4	Test activities, testware, tester-roles	X	X		X	X	X
CTFL	1.5	Essential skills and good practices in testi	X	X	X	X	X	X
CTFL	2.1	Testing in the context of the SDLC	X	X	X	X		
CTFL	2.2	Test levels and test types	X	X	X	X		
CTFL	2.3	Maintenance testing	X	X	X	X		
CTFL	3.1	Static testing basics	X	X	X			X
CTFL	3.2	Feedback and Review process	X	X	X			X
CTFL	4.1	Test techniques overview	X	X	X	X		
CTFL	4.2	Black-Box testing	X	X	X			X
CTFL	4.3	White box testing techniques	X	X	X			

We examined the following syllabi for their impact on economic, technical and environmental sustainability:

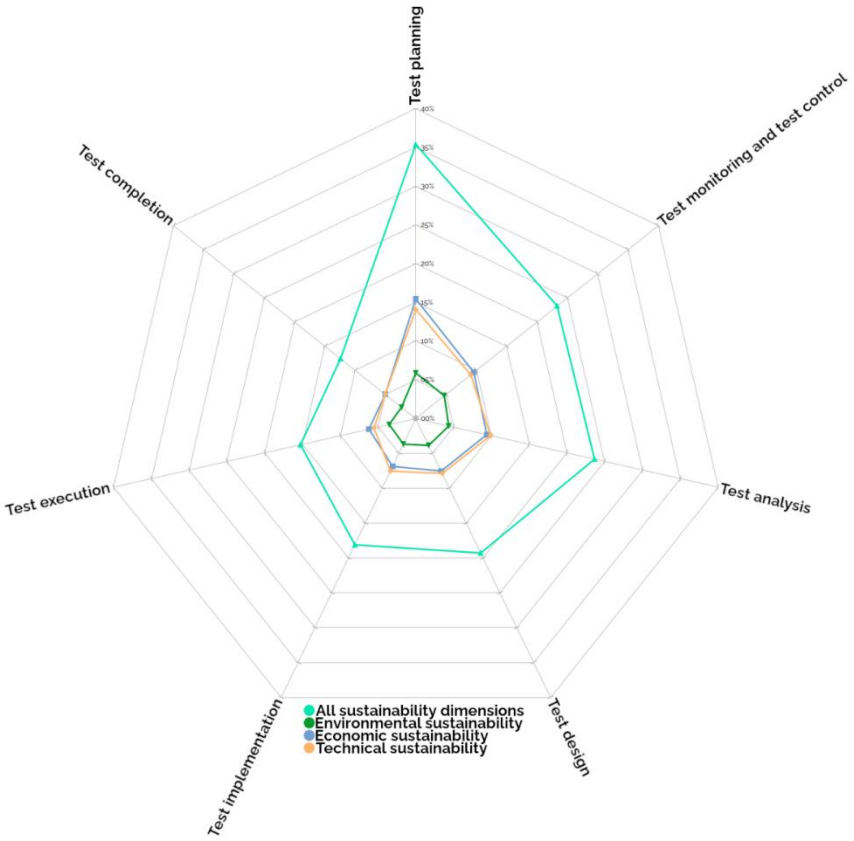
- Certified Tester Foundation Level (CTFL) V4.0

- Certified Tester Advanced Level Test Management (CTAL-TM) V3.0
- Certified Tester Advanced Level Technical Test Analyst (CTAL-TTA) V4.0
- Certified Tester Advanced Level Test Automation Engineer (CTAL-TAE) V2.0
- Certified Tester AI Testing (CT-AI) V1.0
- Certified Tester Testing with Generative AI (CT-GenAI) V1.0
- Certified Tester Model-based Tester (CT-MBT) V1.1
- Certified Tester Performance Testing (CT-PT) V1.0
- Certified Tester Test Automation Strategy (CT-TAS) V1.0

To illustrate the influence of a syllabus on a sustainability dimension, we compared the number of subchapters relevant to a particular dimension with the total number of subchapters. If all chapters of a syllabus were relevant to a sustainability dimension, we obtained a value of 100%.

2.3 Overall Results

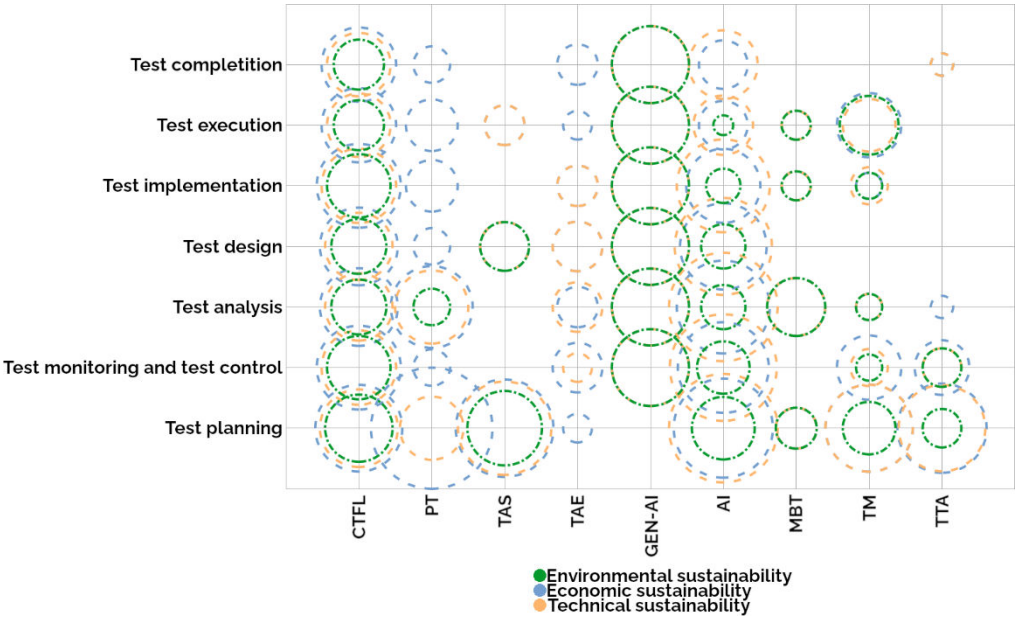
The overview picture shows that test planning is the crucial phase for sustainability. While it is also possible to improve sustainability in later phases, this might be more difficult if the wrong decisions were made at the beginning of the project.



Aggregated Results of all examined Syllabi

The evaluation also shows that there is a close connection between economic and environmental sustainability. This seems logical, since if fewer resources are used, the cost will ultimately be decreased as well.

The differences between the syllabi become clearer when we illustrate their influence on sustainability in a bubble chart:



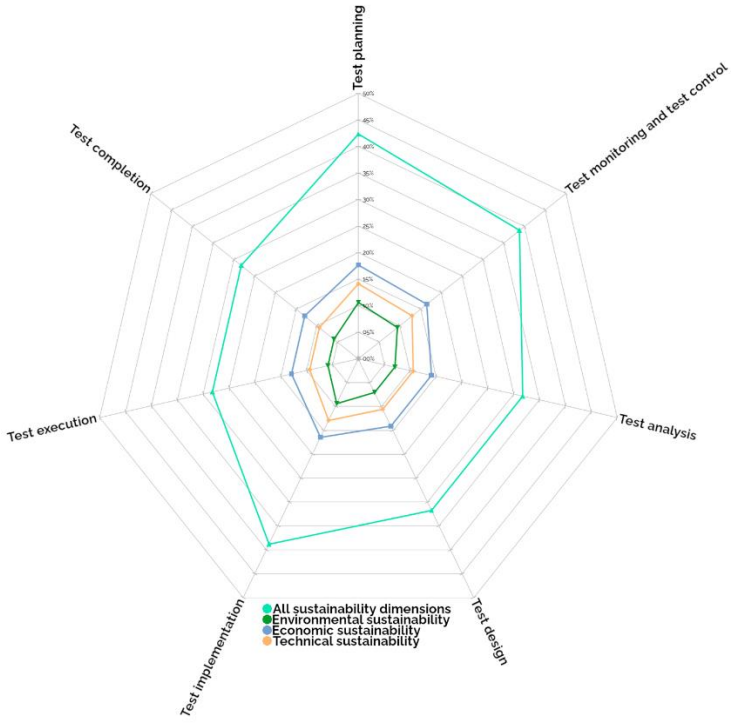
Results of all examined Syllabi (Bubble Chart)

As we can see, the CTFL syllabus offers the most guidance on sustainable testing across all dimensions. Also, the test planning phase has the most to offer across all syllabi.

The CT-AI syllabus stands out in that it has a similar influence in all phases. This is due to the fact that AI can be used in all areas. Depending on how AI is used, it can have a strong positive or negative impact on sustainability.

2.4 Results for CTFL

The CTFL syllabus is of particular importance since it scores highly in all examined sustainability dimensions.



Results of the Certified Tester Foundation Level (CTFL)

Looking at the CTFL syllabus from an environmental sustainability perspective, some basic recommendations can be derived:

- By using systematic test design techniques, we can avoid unnecessary test runs. The use of equivalence classes and boundary value analysis means that only representative inputs are tested, reducing computing time and thus the energy requirements of test environments.
- Another principle is early testing: if defects are detected in early in the project, no extensive code changes or redeployments are necessary.
- Risk-based testing concepts allow test activities to be focused on the most critical functions. By testing less relevant areas less intensively, the overall duration of test cycles can be shortened, thereby minimizing the energy consumption of test infrastructures.

Through these systematic practices, a sound understanding of testing not only contributes to higher software quality, but also actively supports the goals of a resource-efficient and sustainable IT environment.

3 Suggestions for Augmenting Sustainability Aspects in ISTQB Syllabi

Sustainability is at this time considered only indirectly in the syllabi. The term is never explicitly mentioned in the curriculum, though there are already many concepts and recommendations that have a positive effect on the various sustainability dimensions. Several important sustainability aspects are still missing and should be included in future versions.

3.1 Certified Tester Foundation Level Syllabus

Naturally, the starting point should be the Certified Tester Foundation Level syllabus since it represents the basis of the ISTQB curriculum and has by far the highest number of certificates. While the candidates shouldn't be encumbered with too much theoretical background, at least the following basic concepts should be included:

- What is sustainability and why should I consider it as a software tester?
- What are the sustainability dimensions most relevant to software testing? (Environmental, Technical, and Economic Sustainability)
- Which measures and decisions throughout the software testing process affect sustainability?

The general concepts should be complemented by concrete recommendations. Many of them are already included, though scattered throughout the various chapters, while others need to be added. The question arises whether a separate chapter for the sustainability recommendations is required. In our opinion, this would create a certain redundancy, so there is no need to change the current syllabus structure. However, the relation of certain key points to sustainability should be mentioned in the respective chapters. Some examples are:

- Test planning involves many decisions that affect testing effort and resource consumption (e.g., test strategy, metrics to be used, planning of test environments including databases).
- When reviewing requirements, user stories, and design documents, testability, resource consumption, and other sustainability aspects should be considered.
- If a significant impact on sustainability is identified during the review, a more detailed assessment should be performed in which a score for the various sustainability dimensions is assigned. This will make it possible to measure sustainability coverage throughout the project. The participants in the assessment should include different stakeholders such as software engineers, business experts, and the test manager.
- The choice of test methods and test coverage is crucial for the testing effort. The goal is maximum risk coverage with minimal testing effort.
- Focusing on essential test cases and avoiding redundant tests saves effort and resources.
- In test design, a modular structure of test steps helps to increase reusability and maintainability.
- Reviews and static analysis can identify error states and risks early on, thus helping to save costs and resources (shift-left).

- Static analysis is also used to evaluate maintainability. Since maintainability has an impact on effort, cost, and the consumption of energy, it strongly affects sustainability.
- Test execution should be prioritized based on risk, test coverage, and changes to the codebase.
- Test automation can reduce the testing effort, but running tests more often than required can waste resources.
- Testing and monitoring of energy consumption helps to determine whether the application and test framework are environmentally friendly. Since processors account for a significant share of power consumption, measuring their load can provide a rough estimate of the power they consume.
- In the test completion phase, potentially reusable test artifacts should be identified.

3.2 Other Relevant Syllabi

Apart from the CTFL, there are several other syllabi that have a strong relationship to sustainability. We suggest that the general recommendations included in the CTFL are further elaborated on in the context of the respective topic.

Certified Tester Advanced Level Test Management (CTAL-TM)

Since test managers are responsible for many decisions that affect sustainability, it is particularly important to raise their awareness for the topic. The choice of test strategy, testing methods, coverage criteria etc. are crucial for a sustainable testing process. On the other hand, the sustainability properties of the SW under test should also be taken into account.

Certified Tester Advanced Level Technical Test Analyst (CTAL-TTA)

A technical tester should check from a white-box perspective whether the system under test is resource-efficient and easily maintainable.

Certified Tester Advanced Level Test Automation Engineering (CTAL-TAE) + Certified Tester Test Automation Strategy (CT-TAS)

Test automation can save time and effort and is thus important for economic sustainability. On the other hand, resource-efficiency needs to be kept in mind when planning the strategy. Redundancies need to be avoided, and tests should only be run if value is added.

Certified Tester AI Testing (CT-AI)

When testing an AI-based system, resource-efficiency should be a major focus that is already considered during system design. Concrete goals and KPIs should be defined and monitored throughout the project.

Certified Tester Testing with Generative AI (CT-GenAI)

As we have pointed out, the use of AI for generating artifacts such as code or test cases can save a lot of time but also consumes vast amounts of energy. This needs to be considered before a decision to use AI for a certain task is made. If AI is used, the energy consumption should be kept as low as possible, e.g., by choosing an adequate model, designing efficient prompts and providing the appropriate context.

Certified Tester Model-based Tester (CT-MBT)

Model-based testing has a particular benefit on sustainability since it significantly improves the maintainability of the test specification. Instead of having to change each test case individually, it is only necessary to change the model at a few points to generate an updated set of test cases or scripts.

Certified Tester Performance Testing (CT-PT)

Performance testing with a large number of simulated users can consume a lot of resources. To ensure sustainability, it is therefore crucial to plan the tests as resource-efficiently as possible. Performance tests should only be executed when necessary and with a number of users that is adequate for the current test goal.

4 UN Sustainable Development Goals

4.1 Overview

In 2015, the United Nations Member States adopted 17 Sustainable Development Goals (SDGs) that cover social, economic, and environmental sustainability. It is important to recognize that an action in one area can affect the outcome of the others.



https://unric.org/en/united-nations-sustainable-development-goals/

While sustainable testing and the ISTQB syllabi are not equally important to all of these goals, they can have direct or indirect influence on the following areas:

- Goal 4: Quality Education
- Goal 7: Affordable and Clean Energy
- Goal 8: Decent Work and Economic Growth
- Goal 9: Industry, Innovation, and Infrastructure
- Goal 13: Climate Action

4.2 Impact of Testing

4.2.1 Goal 4: Quality Education

SDG 4 aims at ensuring inclusive and equitable quality education and promoting lifelong learning opportunities for all.

The ISTQB syllabi in general support goals 4.3 (access to affordable and quality technical education) and 4.4 (increase number of adults with relevant technical skills). They also have a positive effect on promoting IT technology in developing countries (goals 4.B and 4.C). The inclusion of sustainability aspects in the syllabi improves the education of testers and helps them to reduce their carbon footprint.

Testing – especially in the context of education, software, and organizational processes can have a meaningful positive impact in several ways:

- Improve the quality of education by ensuring the quality of digital learning platforms, educational software, and assessment systems
- Promoting equal access and inclusion by verifying accessibility features and reducing bias in algorithms and digital assessments

4.2.2 Goal 7: Affordable and Clean Energy

SDG 7 aims at ensuring access to affordable, reliable, sustainable and modern energy.

As shown in chapter 1.2, the ICT sector has a significant impact on energy consumption. Several ISTQB syllabi have a considerable effect on goal 7.3 (improving energy efficiency), both by using less energy for SW testing and by building more energy-efficient SW.

- CTFL: Many testing activities have an effect on energy consumption (test strategy, systematic test methods, test automation, reusability and maintainability of test artifacts etc.).
- CT-PT: Performance testing uses a lot of bandwidth, energy, and other resources. Therefore, it is important to consider resource-efficiency when planning the performance test strategy.
- CT-TAS and CTAL-TAE: Test automation can save energy, e.g., by reducing the time for test execution. On the other hand, a poor test automation strategy can also waste energy, e.g., by repeating irrelevant test cases in every single test run.
- CT-AI: AI is using enormous amounts of energy. Therefore, building energy-efficient AI systems has to be a focus point of AI testing.
- CT-GenAI: Using AI for testing is problematic due to its high energy consumption. However, it also has the potential for saving energy by speeding up time-intensive tasks such as test case creation or maintenance.
- CT-MBT: Model-based testing is particularly beneficial for reducing the effort for test case maintenance
- CTAL-TA and CTAL-TTA: Applying systematic test methods rather than redundant or random testing improves testing efficiency

- CTAL-TM: The decisions of the test manager affect all of the aforementioned areas. Therefore, the test manager should always be aware of the effects on energy efficiency.

4.2.3 Goal 8: Decent Work and Economic Growth

SDG 8 aims at promoting sustained, inclusive and sustainable economic growth, full and productive employment, and decent work for all.

Goals 8.2 (higher level of economic productivity) and 8.4 (improve global resource efficiency) are supported by most ISQTB syllabi (e.g., CT-TAS, CT-GenAI, CT-MBT). The ISTQB program also has a positive effect on helping people to get education and find decent jobs (goals 8.5 and 8.6).

Testing can positively influence this goal across economic, social, and organizational dimensions:

- Improve product quality and economic stability
- Support innovation and digital transformation
- Create and preserve high-quality jobs
- Ensure ethical and responsible use of technology

4.2.4 Goal 9: Industry, Innovation, and Infrastructure

SDG 9 focuses on building resilient infrastructure, promoting inclusive and sustainable industrialization, and fostering innovation.

As already explained in goals 4 and 7, the ISQTB syllabi can improve the technical capabilities of developing countries and promote resource-use efficiency (goals 9.4 and 9.5)

Testing plays a key role in supporting this goal by ensuring that technological progress is reliable, resilient, and responsible:

- Build resilient and reliable infrastructure
- Enabling sustainable innovation by integrating testing early and continuously
- Support sustainable industrialization by detecting inefficiencies in production and digital processes
- Ensure long-term maintainability of systems
- Ensure and improve quality and safety standards

4.2.5 Goal 13: Climate Action

SDG 13 focuses on taking urgent action to combat climate change and its impacts.

As already explained in goal 7 and our analysis in chapter 2.3, testing can have a huge impact on reducing greenhouse gas emissions through improved energy efficiency. Also, if sustainability aspects are included in the ISTQB syllabi, this would improve education and raise awareness on climate change issues.

5 Sustainability vs. AI

The list of tasks performed by Artificial Intelligence (AI) is rapidly growing, both in our daily lives and in IT projects. Why write the code yourself if it can be generated? Why tax your brain for test cases if they pop out of the GenAI platform? With faster test case creation, smarter prioritization, and improved defect prediction, AI promises greater efficiency in software testing.

At the same time, the use of AI places new demands on data, computing power, infrastructure and people. This leads to conflicts of interest where economic, ecological, and social sustainability goals collide. If we want to reach the UN Sustainable Development Goals and keep our planet habitable, should we really delegate every minor task to AI?

In this chapter, we point out possible conflicts of interest that should be considered when deciding about the use of AI for software testing.



This is how AI depicts the area of conflict "Mother Nature vs. AI"

5.1 Conflicts

While the economic view mainly focuses on cost, time-to-market, and risk reduction, the ecological perspective is on resource consumption (hardware and energy production, infrastructure operation, and all resulting emissions along the entire production chain) beyond its immediate monetary cost. Employees are confronted with new technology and the skill changes made necessary thereby. The following conflicts should be addressed:

5.1.1 Energy and emissions profile comparison

Training large models is the most visible source of emissions, but in a test context this often takes place less frequently than inference. Inference load, especially in CI/CD with frequent pipeline runs, though less energy-intensive than training, still rapidly adds up

over recurring test cycles. Traditional automation without AI (rule-based orchestration, risk-based testing, selection of test values) usually has a lower and more predictable energy profile. The key point is the comparison of "benefit per kWh": Does the use of AI deliver measurably higher error detection rates or relevant time savings that justify the additional energy profile? Without this relationship, the additional ecological demand remains difficult to justify.

5.1.2 Data conflicts: scope, quality, and retention

AI thrives on the diversity and volume of its training data, while sustainability requires data minimization, compliance and short retention periods. AI-generated synthetic test data can replace real, sensitive data, reducing risks and often storage requirements, provided that the test objectives are sufficiently specified when prompting its generation. Data quality is another source of conflict, better quality increases AI usefulness, but often at the cost of time-consuming preparation. In addition, long retention periods exacerbate the resource burden. A conscious approach to sampling, anonymization, deduplication, and clear deletion periods is therefore essential.

5.1.3 Infrastructure conflicts: location, hardware, operating model

The emission factor of a data center depends heavily on the regional electricity mix. Locations with a high proportion of renewable energies reduce the ecological footprint, while regions with energy production from predominantly non-renewable sources exacerbate the conflict. Specialized hardware (GPU/TPU) for the test environment increases efficiency per task, but its manufacture causes "embodied emissions" that must be included in the overall calculation. Cloud computing offers flexible scaling and often better energy efficiency, while on-premise solutions provide control but may be disadvantageous if the infrastructure is outdated. This conflict can be addressed at the organizational level with operating models such as "low-carbon scheduling" (testing at times when the share of renewable energy is high) or "GreenOps" (targeted resource management).

5.1.4 Rebound and scaling effects

Where AI makes testing faster and cheaper, the test frequency or coverage often rises. This will increase overall emissions, even if the individual test runs more efficiently. These rebound effects are particularly relevant in everyday testing: More nightly builds, additional regressions, and broader data coverage are attractive from a quality assurance perspective but drive up resource requirements. Attainable sustainability targets may be to introduce limits such as emission budgets per release, capping the test frequency for marginal increases in benefit, or clear thresholds above which additional test runs are only performed if there is a considerable risk criticality. Our analysis in chapter 2.3 shows the importance of test planning for this topic.

5.1.5 Social impact

The use of AI always has an impact on people and organizations. Qualification requirements shift, pressure to upskill arises, roles may be eliminated, and inequalities within teams are exacerbated. Test managers must be able to evaluate the significance of the non-deterministic results of AI tests and communicate this to their stakeholders.

Test data practices (collection, processing, storage) touch on privacy, co-determination, and trust when production data is used. Non-transparent models might blur responsibilities, reproduce bias, and unfairly influence quality judgments. In addition, power asymmetries arise from vendor lock-in or centralized AI competencies, which make collaboration difficult and reduce acceptance. These tensions should be addressed through clear governance, ethical guidelines, transparency, and employee inclusion before and during implementation.

5.2 Conclusion

The conflict between AI use and sustainability in software testing is real and may not be easily resolved in some areas. It becomes manageable when benefits and costs are made measurable, explicit limits are set, and technical and organizational guidance is applied consistently.

6 Final Thoughts and Future Work

As we have seen, the ICT sector has a huge impact on our capability to reach the UN Sustainable Development Goals. In particular, the ever-increasing use of AI provides new opportunities but also creates a number of conflicts with sustainability interests.

These points should also be considered in software testing. While the current versions of the ISTQB syllabi do not explicitly mention sustainability, they already contain numerous chapters related to the topic and hints that help to make testing more sustainable. In our opinion, enhancing the visibility of sustainability aspects and putting more emphasis on the topic in the relevant syllabi would help testers to implement concrete sustainability measures in their daily work. It would also accentuate ISTQB's commitment to responsibly maintaining our planet for future generations.

Possible next steps to continue our work include:

- Participation in improving sustainability aspects in existing ISTQB syllabi
- Creation of a new syllabus specifically dedicated to sustainable testing and/or testing sustainability
- Promoting the topic through conference presentations and journal articles
- Cooperation with other organizations promoting sustainability

7 References

[1] G. Brundtland, "Our Common Future (Brundtland Report)," United Nations World Commission on Environment and Development, 1987.

[2] B. Penzenstadler and H. Femmer, "A generic model for sustainability with process- and product-specific instances," in *Proceedings of the 2013 Workshop on Green in/by Software Engineering*, 2013.

[3] S. Naumann, M. Dick, E. Kern and T. Johann, "The GREENSOFT Model: A reference model for green and sustainable software," *Sustainable Computing: Informatics and Systems,* pp. 294-304, 2011.