Department of
Computers and Informatics
FEEI TU of Košice

Modern Software Evolution: The Path from Working Software to Green Software

Csaba Szabó
Technical university of Košice, Slovakia
csaba.szabo@tuke.sk

GREEN 2022, October 16-20, 2022
Lisbon, Portugal

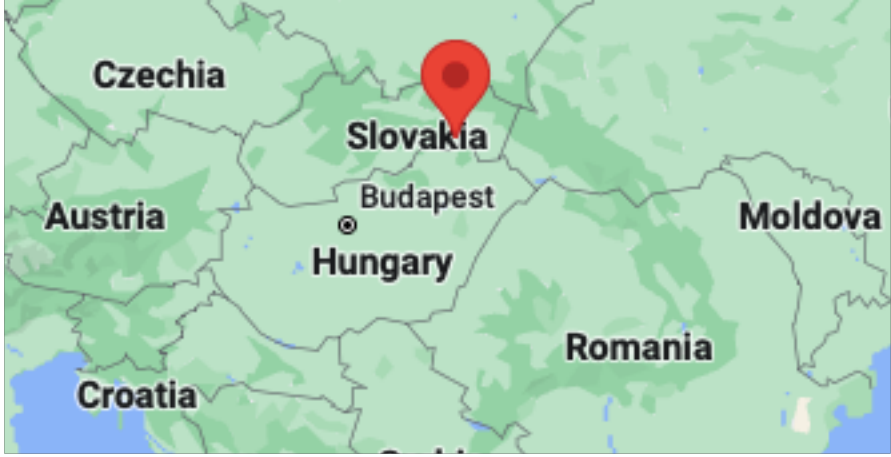
IARIA

About the Author

Csaba Szabó completed his MSc. and PhD. studies and habilitated in Computer Science at the Faculty of Electrical Engineering and Informatics, Technical University of Košice, Slovakia, where he is currently working as Associate Professor at the Department of Computers and Informatics. During his PhD studies, he spent one semester at Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary, which also included solving of technical tasks for the local branch of the ALSTOM company. He also completed short term studies in Szeged, Hungary and Subotica, Serbia.

https://kpi.fei.tuke.sk/sk/person/csaba-szabo

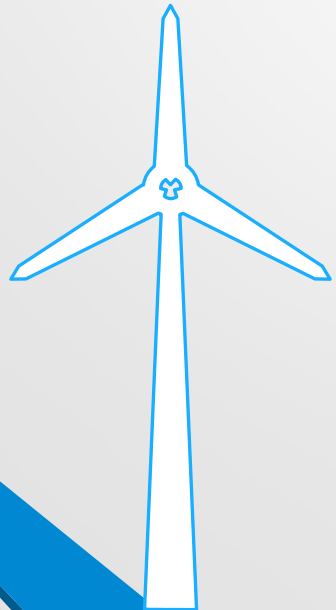
About the Author

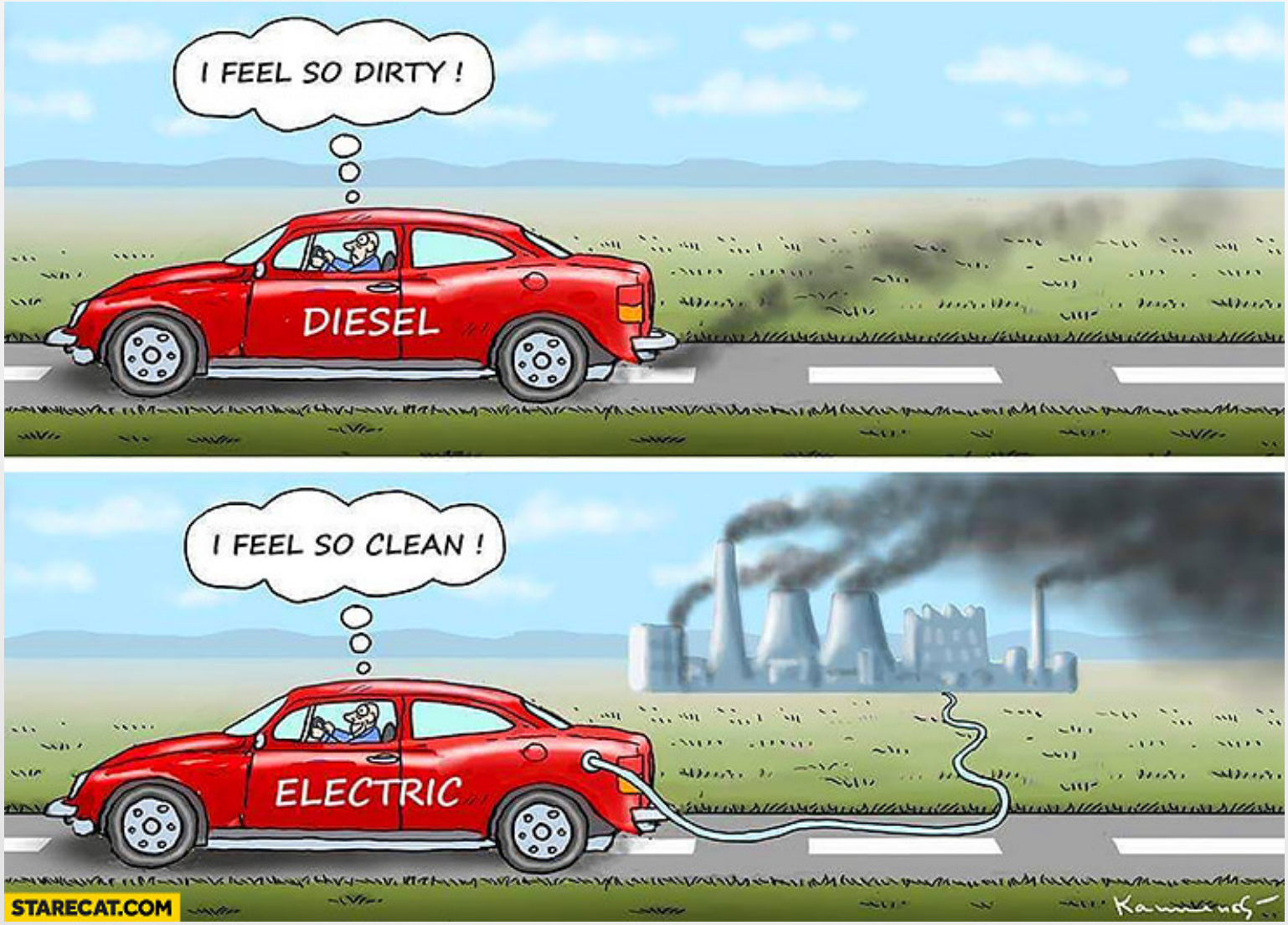


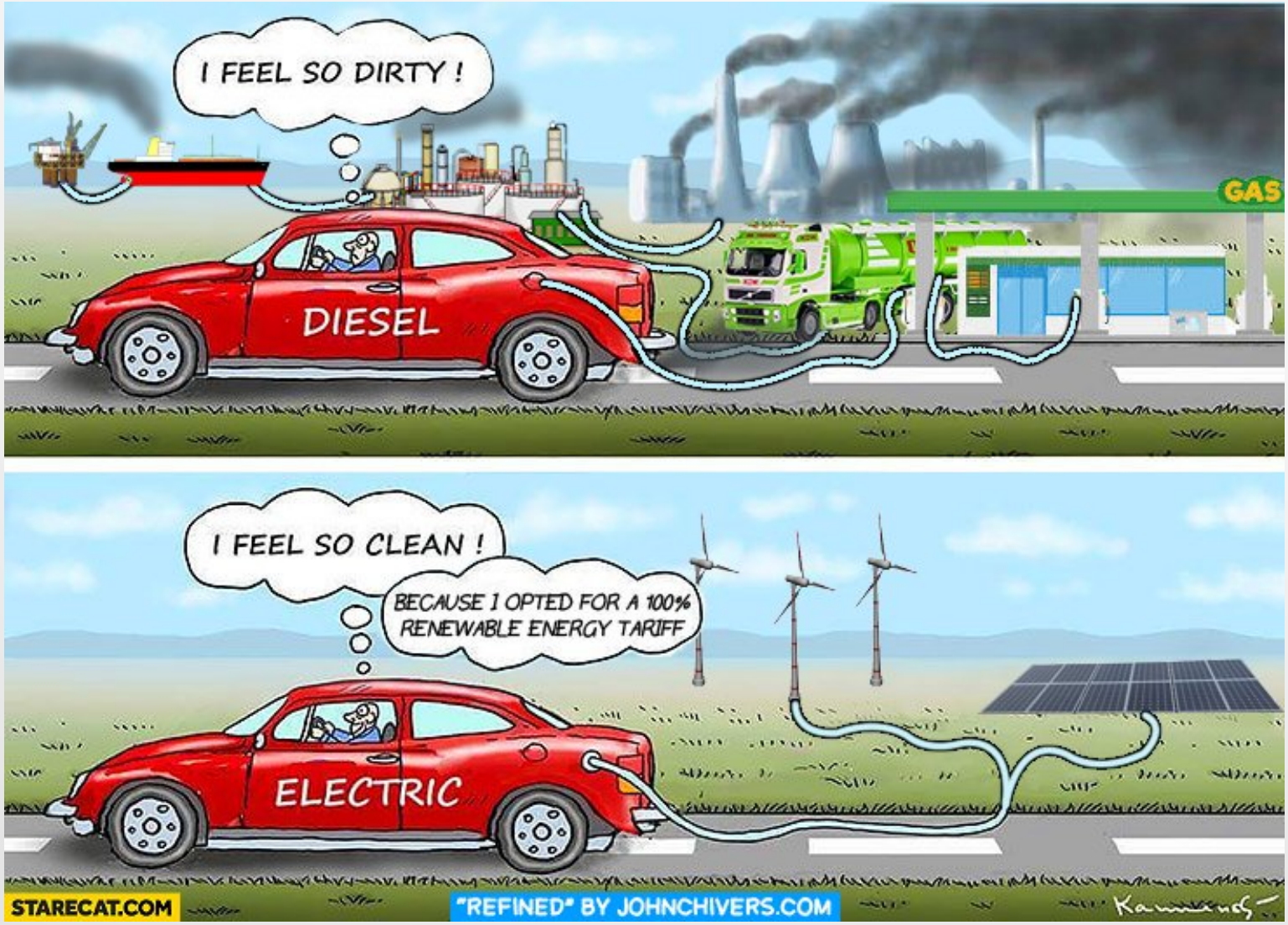
He is involved in different research areas, mainly in green software, software evolution, software project management, software testing and virtual reality. In these fields, he has international cooperation with researchers from numerous European countries (AT, BG, HR, HU, NL, PT, RO, RS) and Egypt. He had invited talks at Mipro 2019 (Opatija, Croatia) and GSERITA 2022 (Virtual event) in the topic of green software. He led the ERASMUS+ project Focusing Education on Composability, Comprehensibility and Correctness of Working Software (2017-1-SK01-KA203-035402), and he is member of the project SusTrainable - Promoting Sustainability as a Fundamental Driver in Software Development Training and Education (2020-1-PT01-KA203-078646). Currently, he is also actively involved and applying his research results in frame of the project Intelligent Systems for UAV Real-Time Operation and Data Processing (ITMS2014+: 313011V422). His research on critical systems, software evolution, testing and management includes a cooperation with R-SYS, subsidiary of ERA.

In this presentation…

- Engineering green software
- Working software
- Software evolution
- Energy consumption estimation
- Energy-inspired evolution game (agile development)




I FEEL SO DIRTY !
DIESEL
I FEEL SO CLEAN !
ELECTRIC
STARECAT.COM

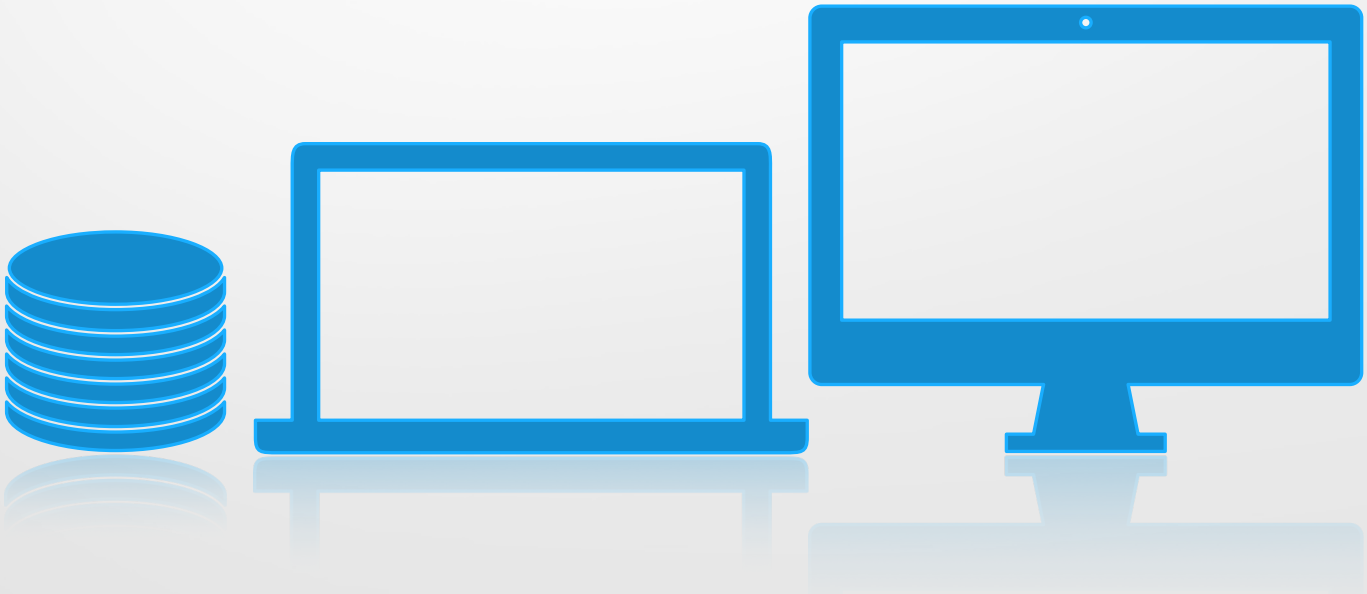

I FEEL SO DIRTY !
GAS
DIESEL
I FEEL SO CLEAN !
BECAUSE I OPTED FOR A 100% RENEWABLE ENERGY TARIFF
ELECTRIC
STARECAT.COM
"REFINED" BY JOHNCHIVERS.COM

Green Software Engineering

Green software engineering is a branch of software engineering focusing on energy aspects of software. Please note that software plays here the role of the process, which one's energy consumption can be expressed through the energy consumption of all hardware parts that are used in any way by the examined process. Therefore, when evaluating software greenness, we always examine the usage load on hardware parts during software execution time.

The role of hardware

- Display
- Networking (Wi-Fi, Bluetooth), radio
- Processor
- Memory
- Disks
- Battery
- Sensors



The role of software

- Operating system (difference between Windows, Linux, macOS, Android, iOS)
- Working software
- Computer games
- Application systems
- Databases

Working Software (WS)

- Working software is a tested software that delivers value to the end-user, value that works well, maybe even better than expected, but never worse.
- Working software is a software which is fully integrated, tested, and ready to be shipped to customers or deployed into production.

* WS is the key measure as defined in the Agile Manifesto
https://agilemanifesto.org/principles.html

** WS definitions by Ekaterina Novoseltseva
https://apiumhub.com/tech-blog-barcelona/working-software-go-live-strategy/

The role of the user

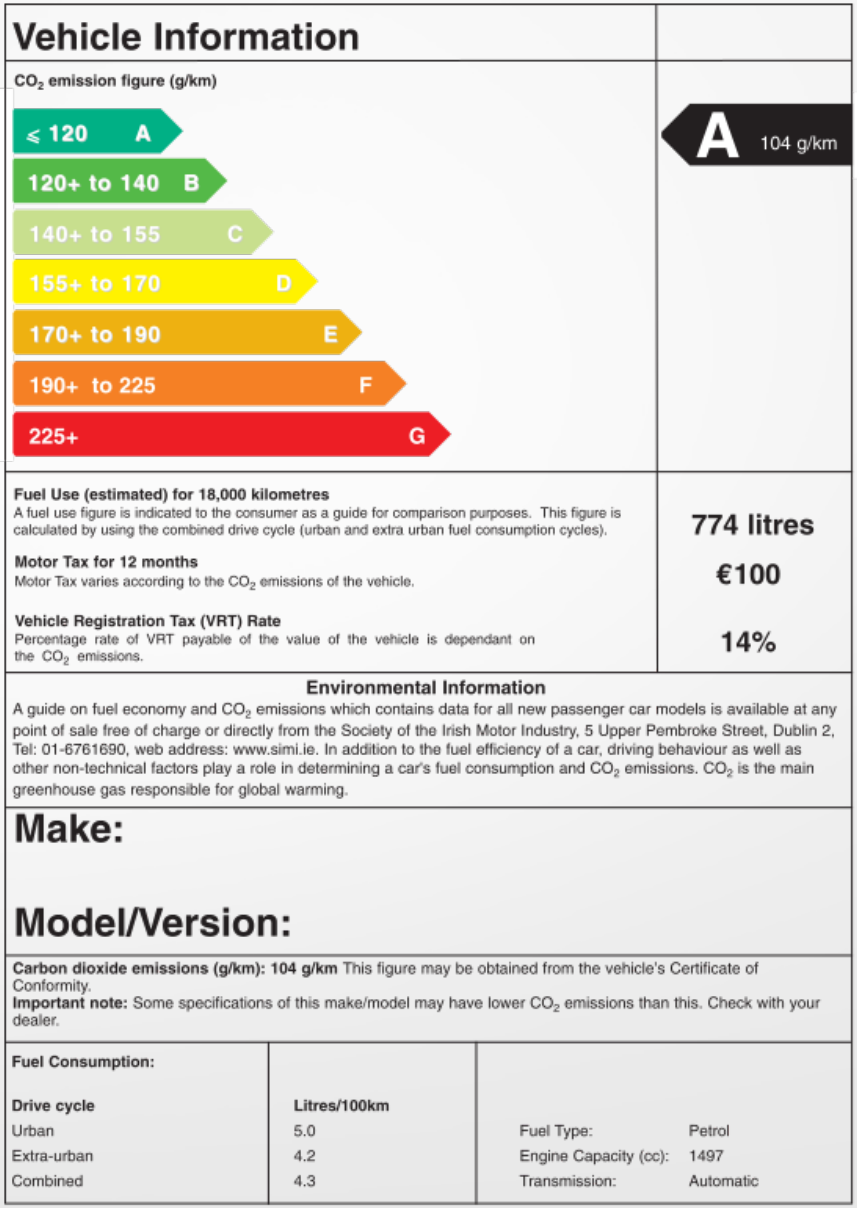
- The user "drives" the software
- Needs individual training (unlike HW/SW)
- Does (s)he receive it? Where?
- Repairing bad configuration is often done by buying a new device… ☹

Green software, green IT

- Goals:
  - Save energy by more efficient hardware
  - Save energy by optimised/custom software
  - Save energy by location of hardware
- To make it really green:
  - Develop new working hardware
  - Develop energy efficient working software
  - Teach users to save energy when using the software
  - Make sure the used energy is also green

SW Energy Label?

HW/SW system energy label?



Vehicle Information

CO_2 emission figure (g/km)

- ≤ 120 A
- 120+ to 140 B
- 140+ to 155 C
- 155+ to 170 D
- 170+ to 190 E
- 190+ to 225 F
- 225+ G

A 104 g/km

Fuel Use (estimated) for 18,000 kilometres — 774 litres
A fuel use figure is indicated to the consumer as a guide for comparison purposes. This figure is calculated by using the combined drive cycle (urban and extra urban fuel consumption cycles).

Motor Tax for 12 months — €100
Motor Tax varies according to the CO_2 emissions of the vehicle.

Vehicle Registration Tax (VRT) Rate — 14%
Percentage rate of VRT payable of the value of the vehicle is dependant on the CO_2 emissions.

Environmental Information

A guide on fuel economy and CO_2 emissions which contains data for all new passenger car models is available at any point of sale free of charge or directly from the Society of the Irish Motor Industry, 5 Upper Pembroke Street, Dublin 2, Tel: 01-6761690, web address: www.simi.ie. In addition to the fuel efficiency of a car, driving behaviour as well as other non-technical factors play a role in determining a car's fuel consumption and CO_2 emissions. CO_2 is the main greenhouse gas responsible for global warming.

Make:

Model/Version:

Carbon dioxide emissions (g/km): 104 g/km This figure may be obtained from the vehicle's Certificate of Conformity.
Important note: Some specifications of this make/model may have lower CO_2 emissions than this. Check with your dealer.

Fuel Consumption:

Drive cycle	Litres/100km
Urban	5.0
Extra-urban	4.2
Combined	4.3

Fuel Type:	Petrol
Engine Capacity (cc):	1497
Transmission:	Automatic

Image from Wikipedia

Measuring energy consumption

Incl. improvements

- System level
- Application level
- Component level
- Code level

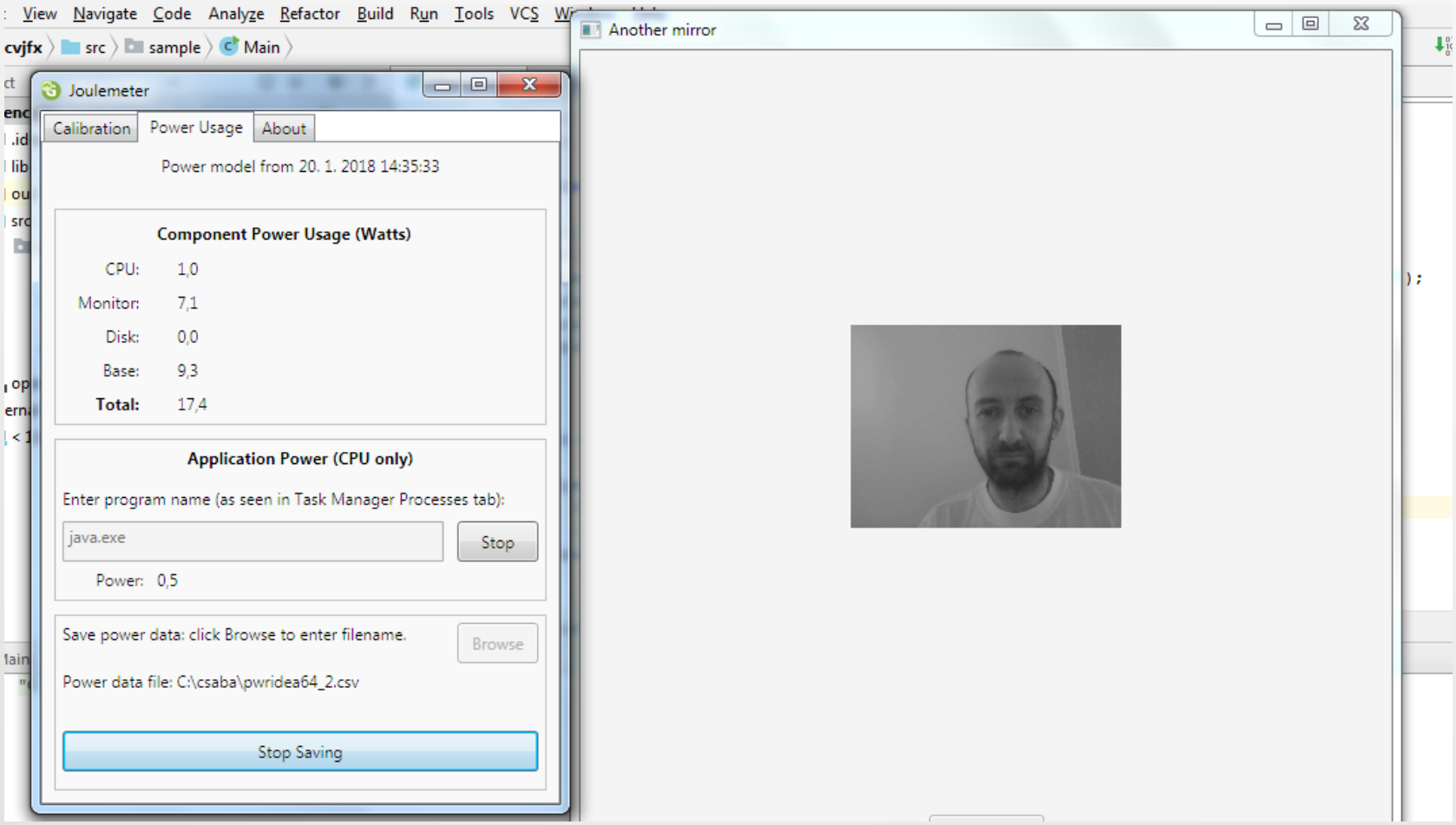
- Process level

More details can be found in Intellectual output 1 of the project 2017-1-SK01-KA203-035402: Focusing Education on Composability, Comprehensibility and Correctness of Working Software

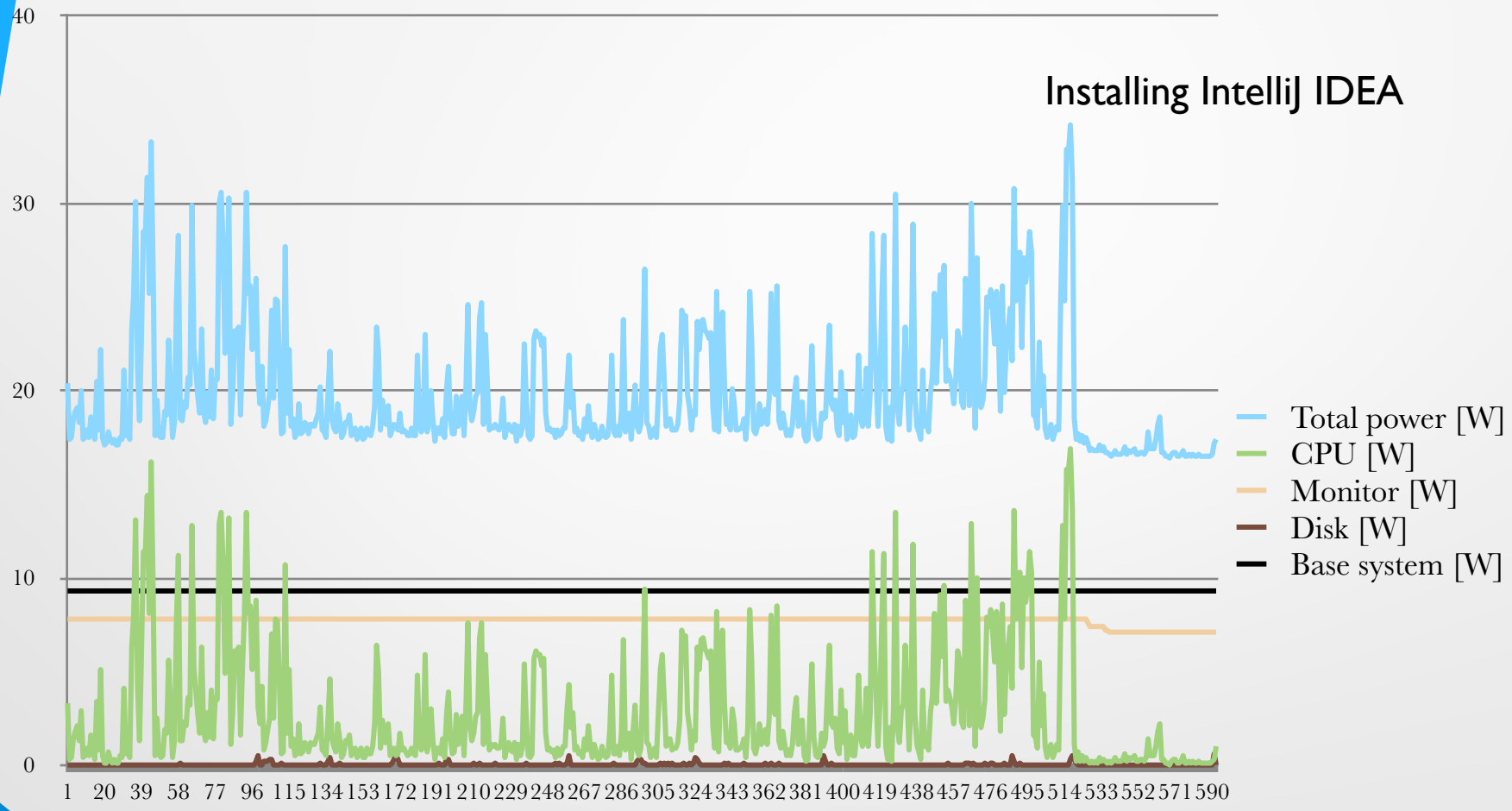
System level measurement

- SW-to-SW/HW solutions (servers, IoT)
- Uptime/availability prediction
- Providing a different evaluation perspective

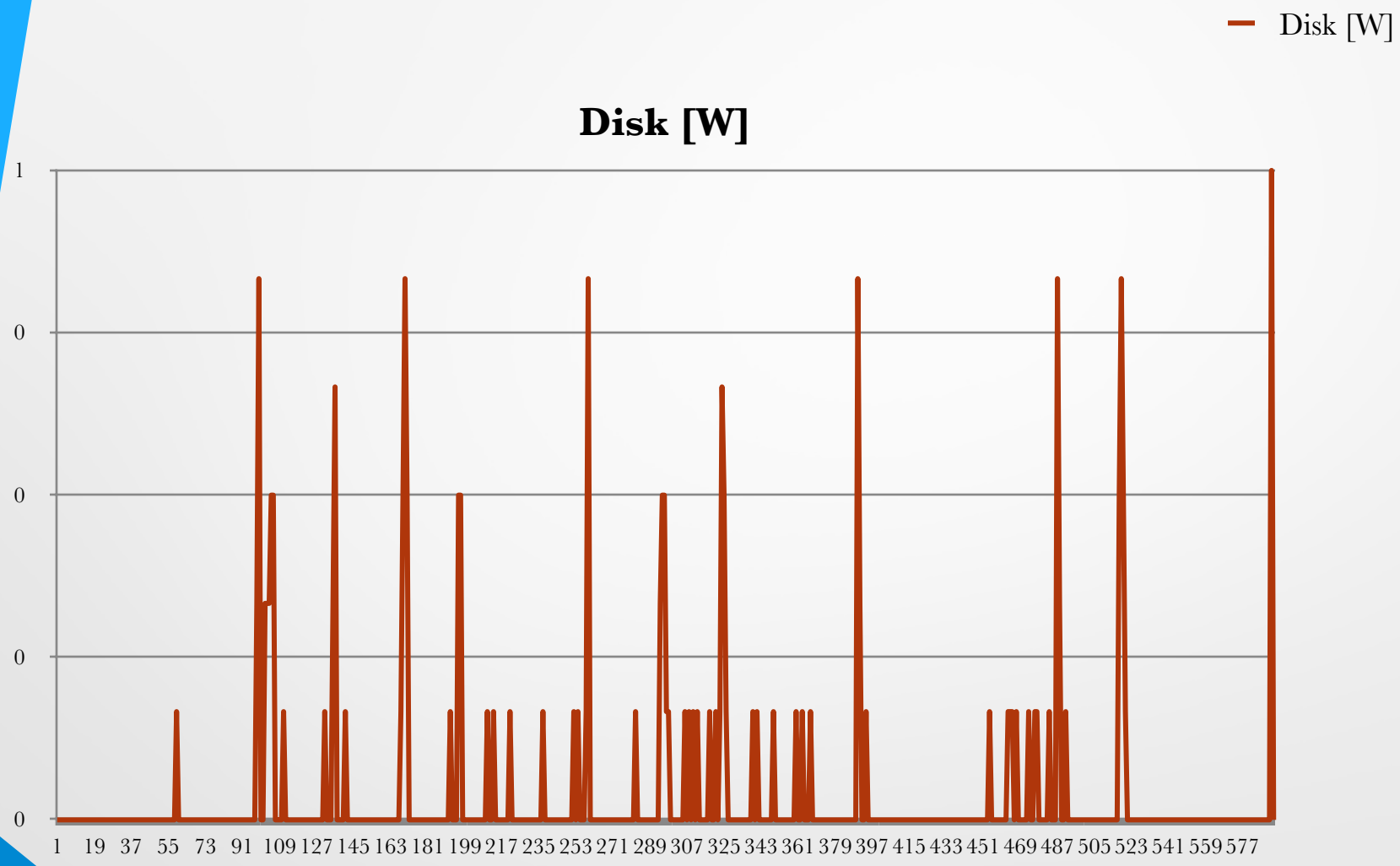
More details can be found in Intellectual output 1 of the project 2017-1-SK01-KA203-035402: Focusing Education on Composability, Comprehensibility and Correctness of Working Software


View Navigate Code Analyze Refactor Build Run Tools VCS
cvjfx src sample Main
Joulemeter
Calibration
Power Usage
About
Power model from 20. 1. 2018 14:35:33
Component Power Usage (Watts)
CPU: 1,0
Monitor: 7,1
Disk: 0,0
Base: 9,3
Total: 17,4
Application Power (CPU only)
Enter program name (as seen in Task Manager Processes tab):
java.exe
Stop
Power: 0,5
Save power data: click Browse to enter filename.
Browse
Power data file: C:\csaba\pwridea64_2.csv
Stop Saving
Another mirror

Application level measurement



More details can be found in Intellectual output 1 of the project 2017-1-SK01-KA203-035402: Focusing Education on Composability, Comprehensibility and Correctness of Working Software



More details can be found in Intellectual output 1 of the project 2017-1-SK01-KA203-035402: Focusing Education on Composability, Comprehensibility and Correctness of Working Software

Component level measurement

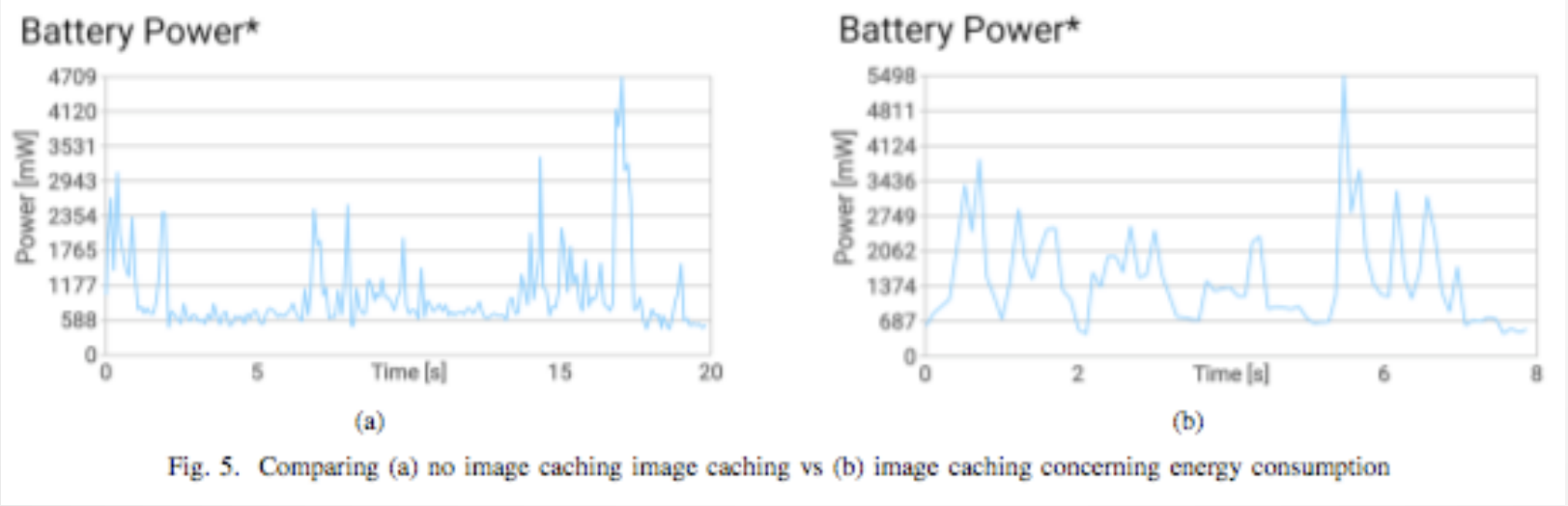


Fig. 5. Comparing (a) no image caching image caching vs (b) image caching concerning energy consumption

- Test oracles
- Comparing different versions
- The driver of energy (r)evolution

More details can be found in Intellectual output 2 of the project 2017-1-SK01-KA203-035402: Focusing Education on Composability, Comprehensibility and Correctness of Working Software

Code level measurement

- Which version of an algorithm is consuming less energy?
- Is it more efficient to store objects in an array than in a list?
- How significantly does the length of execution impact on the consumption measured when generating file MD5/SHA-n values?

The energy-measured development game

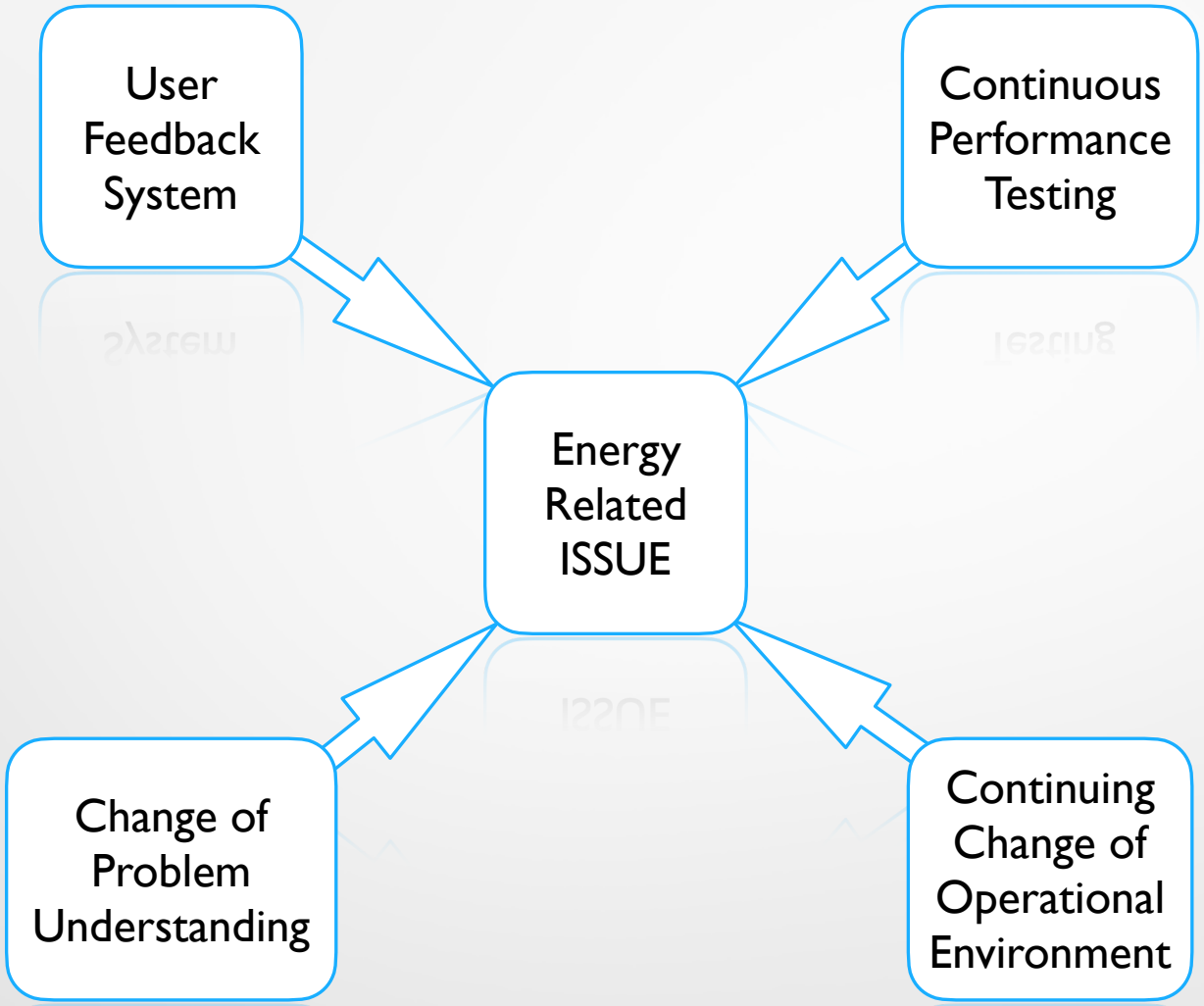
1. Setup the environment
2. Start the energy monitor
3. Develop (think, code, test, fix) for 15 minutes
4. Have a 5 minutes break (stop energy usage monitoring, set up the next one, get a coffee)
5. Finish (for this time) if there is no further idea
6. Repeat (jump to label 2)
7. Analyse collected data (energy efficiency of your development process) inside the team

More details can be found in Intellectual output 1 and 2 of the project 2017-1-SK01-KA203-035402: Focusing Education on Composability, Comprehensibility and Correctness of Working Software

Software Evolution

- S/P/E-type software (Lehman, 1980 and later):
  - Continuing Change
  - Increasing Complexity
  - Declining Quality
  - Feedback System
- Agile development is evolutionary development (WS is P/E-type software — more E-type, less P-type)

Lehman, M. M. (1980). "Programs, Life Cycles, and Laws of Software Evolution". Proc. IEEE. 68 (9): 1060–1076. doi:10.1109/proc.1980.11805


User Feedback System
Continuous Performance Testing
Energy Related ISSUE
Change of Problem Understanding
Continuing Change of Operational Environment

The energy-inspired evolution game

1. Setup the environment
2. Start the energy monitor
3. Identify an energy leak, reconstruct an existing issue
4. Have a 5 minutes break (stop energy usage monitoring, get a coffee)
5. Evolve the code based on the energy leak or issue
6. Finish (for this time) if there is no further idea
7. Repeat (jump to label 2)
8. Release new version of WS

GSE

The future of

- Research:
  - Unification of principles
  - Standardisation of representation of results
  - Software energy efficiency (label?)
  - Sustainable AI
- Industry:
  - Sustainable AI in software
  - Sustainable autonomous robots
- Academia:
  - Prepare this future by educating the people

TECHNICAL UNIVERSITY OF KOŠICE
Faculty of Electrical Engineering and Informatics

Liked it? ☺

Csaba.Szabo@tuke.sk

Disclaimer

The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Union. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

This presentation contains parts of Intellectual output 1 and Intellectual output 2 of the ERASMUS+ project No. 2017-1-SK01-KA203-035402: Focusing Education on Composability, Comprehensibility and Correctness of Working Software.

Co-funded by the
Erasmus+ Programme
of the European Union