

Kestävä ohjelmistokehitys ja vihreä koodaus

Laura Partanen, FT, Tutkija, LUT-yliopisto



**Euroopan unionin
osarahoittama**

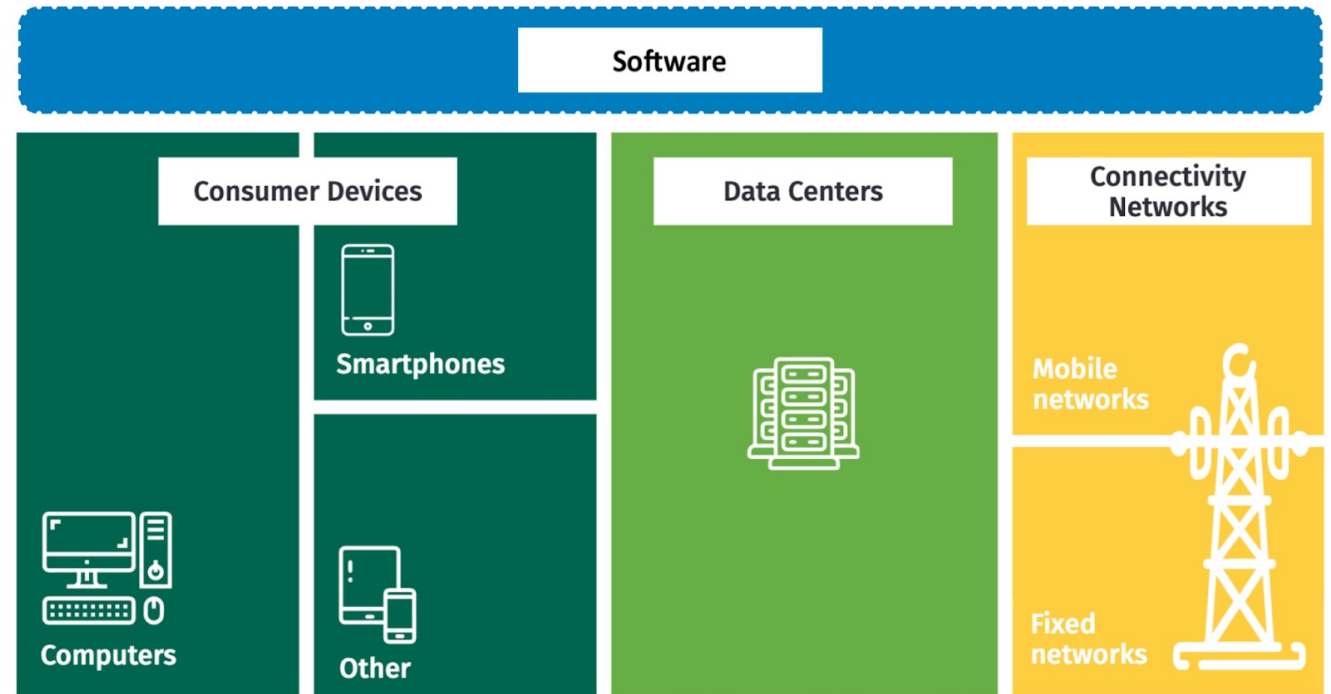


Elinvoimakeskus



Miksi kestävä ohjelmistokehitys on tärkeää?

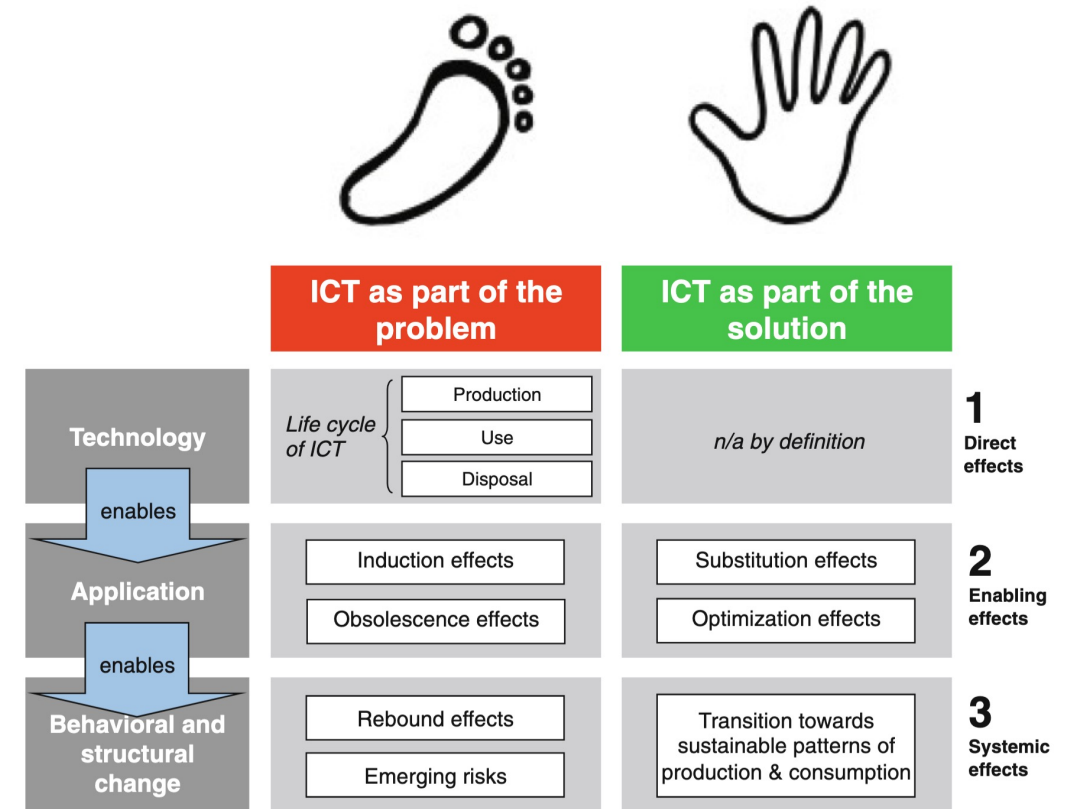
- Digitalisaatiolla on yhä merkityksellisempi asema yhteiskunnassa.
- Pienet tehottomuudet skaalautuvat valtaviksi, kun käyttäjämäärät ovat suuria.
- Kestävyys on osa ohjelmiston laatua.



Partanen, L., Sipilä, A., Haque, Md S. & Porras, J. Mapping of the system of software related emissions and shared responsibilities. Based on Ayers, Seth; Ballan, Sara; Gray, Vanessa; McDonald, Rosie. *Measuring the Emissions and Energy Footprint of the ICT Sector : Implications for Climate Action* ((English). Washington, D.C. : World Bank Group.

Vaikutusten aikaskaala

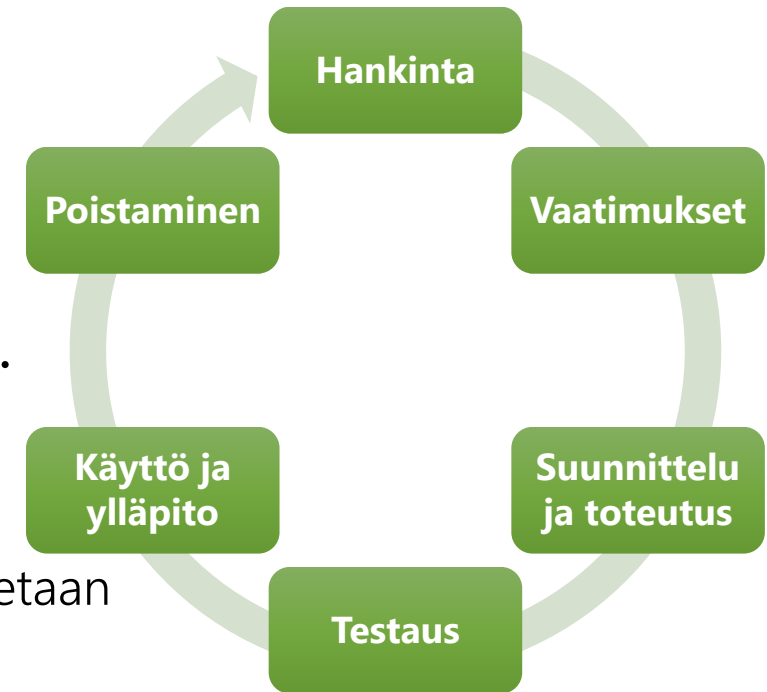
- Vaikutukset voivat tapahtua kolmella eri aikaskaalalla (orders of impact)
 - **Suorat vaikutukset** - kun vaikutus on välitön tarkasteltavan kohteen suhteen. Esim. tietokoneen elinkaari: valmistus, käyttö ja käytöstä poisto
 - **Epäsuorat vaikutukset** - kun vaikutus seuraa tarkasteltavan kohteen käytöstä. Esim. tietokoneen ja ICT-infrastruktuurin avulla voidaan lähettää e-lasku paperilaskun sijasta
 - **Rakenteelliset vaikutukset** - kun vaikutus johtaa käytöksen muutokseen yksilön tai yhteisön tasolla. Esim. e-laskun käyttö johtaa kirjepostin vähenemiseen ja sitä kautta postinjakeluun.



Hilty L. and Aabischer B.: ICT Innovations for Sustainability, Springer, 2013

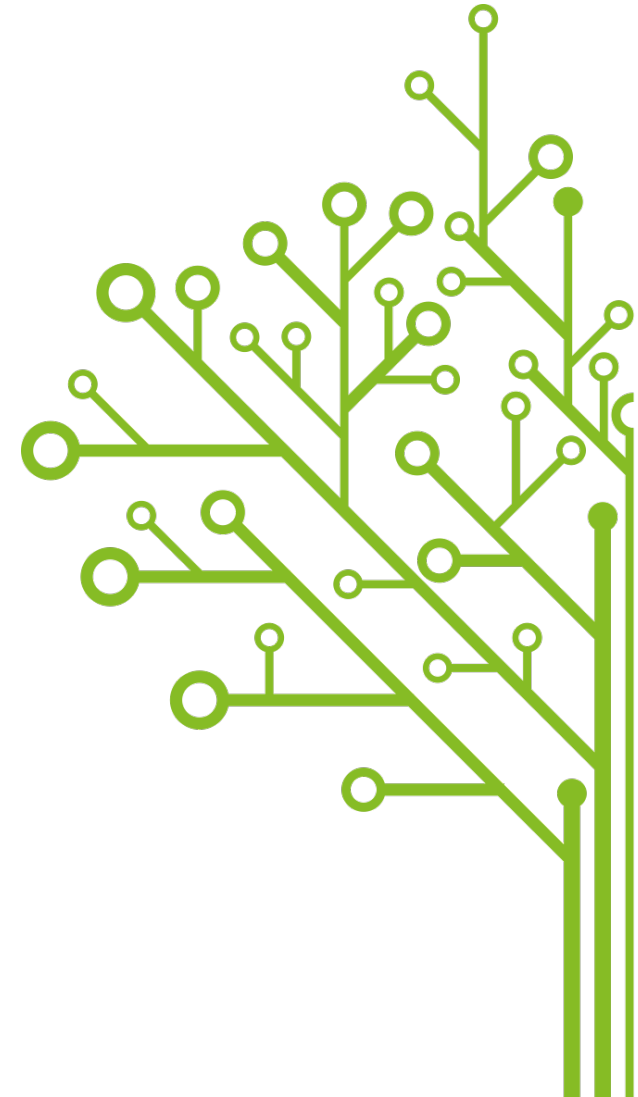
Kestävä ohjelmistokehitys elinkaaren näkökulmasta

- Kestävyys syntyy päätöksistä jokaisessa vaiheessa – se ei ole loppuvaiheen optimointi.
- Kestävyys huomioidaan konkreettisesti jokaisessa elinkaaren vaiheessa:
 - **Hankinta:** toimittajilta avoimuutta energiankulutuksesta, tunnistetaan kestävyysvaikutuksista
 - **Vaatimukset:** dataminimalismi ja energiatehokkuus
 - **Suunnittelu ja toteutus:** kevyet ratkaisut, tehokkaat algoritmit
 - **Testaus:** kohdennettu testaus, resurssioppien tunnistaminen
 - **Käyttö ja ylläpito:** jatkuva optimointi, tarpeettoman kapasiteetin poisto
 - **Poistaminen:** hallittu alasajo, datan karsinta ja siivous



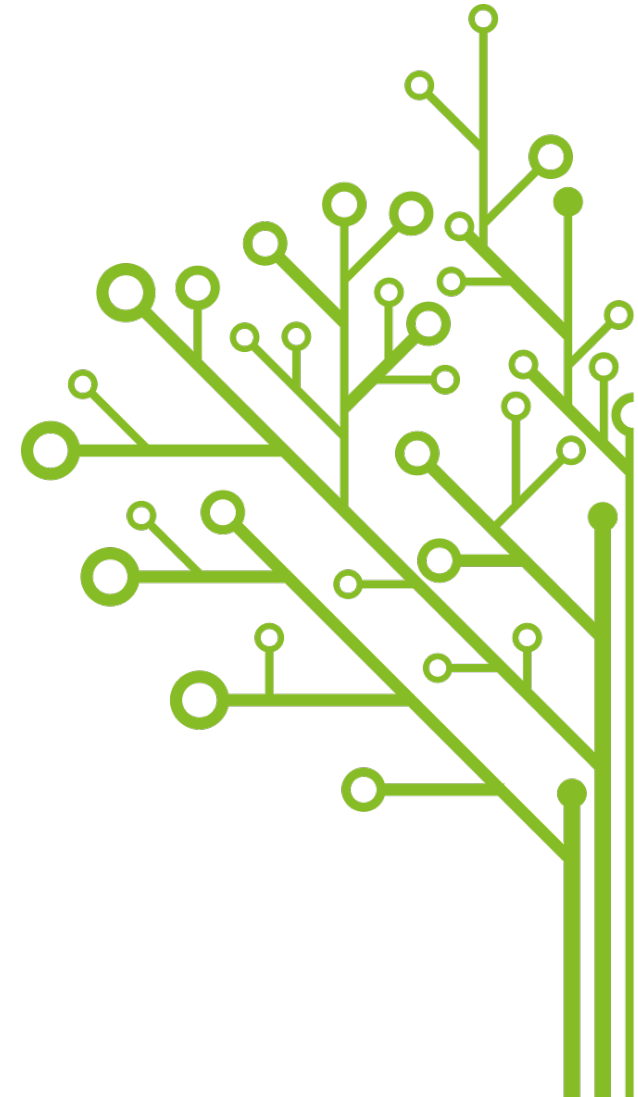
Mitä on vihreä koodaus?

- Ohjelmiston suunnittelua ja toteutusta siten, että vältetään tarpeetonta ajonaikaista energian- ja resurssienkulutusta ohjelmiston toiminnallisuus ja laatu säilyttäen.
 - Fokus on ajonaikaisessa energiankulutuksessa
 - Käyttäytyy työkuorman mukaan
 - Määräytyy koodin ja arkkitehtuurivalintojen kautta
 - Mitattavaa ja dataan perustuvaa



Mitä vihreä koodaus ei ole (yleisiä väärinkäsityksiä)

- Ei pelkkää suorituskyvyn optimointia.
- Ei kertaluonteinen tehtävä.
- Ei ennenaikaista mikrotason optimointia.
- Ei ylläpidettävyyden uhraamista.
- **Vihreässä koodauksessa on kyse tietoisista kompromisseista.**



Käytännön askeleita aloittamiseen

1

Tunnista järjestelmän
energiaintensiiviset osat

2

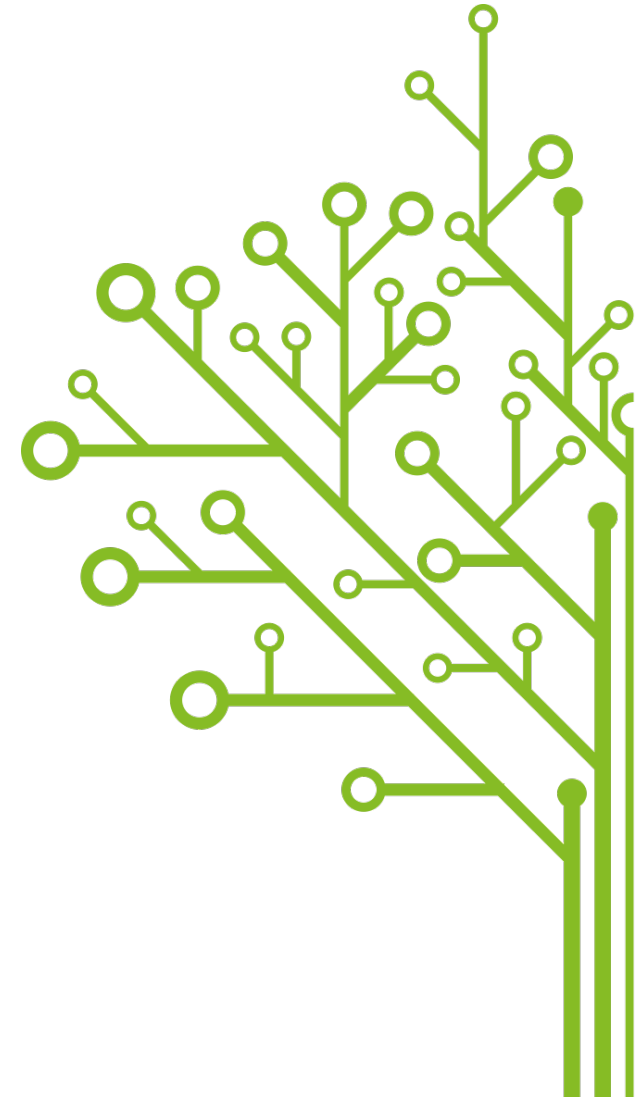
Sovella domain-kohtaisia
green coding -käytäntöjä

3

Käytä mittaamista ja vertailua
jatkuvaan parantamiseen

Vaihe 1: Tunnista energiaintensiiviset osat

- Usein suoritettavat koodin osat
- Paljon liikennettä käsittelevät API-rajapinnat
- Pitkään käyvät taustatehtävät
- Suuria datamääriä käsittelevät prosessit
- Tekoälyn koulutus- tai ennustetehtävät



Vaihe 2: Sovella domain-kohtaisia käytäntöjä

Frontend

- Vähennä uudelleenrenderöintejä.
- Pienennä bundle-kokoa.
- Vältä turhan tiheää API-kutsumista.

Backend

- Vältä N+1-kyselyitä.
- Vähennä työtä per pyyntö.
- Käytä rajattua ja hallittua välimuistia.

Tekoäly

- Valitse käyttötarkoitukseen sopivan kokoinen malli.
- Pienennä promptien ja tokenien määrää.
- Vältä turhia uudelleenkoulutuksia.

Infrastruktuuri

- Mitoita laskentaresurssit oikein.
- Vältä ylivoimista.
- Hyödynnä automaattista skaalausta ja suorituskykyä tukevia konfiguraatioita.

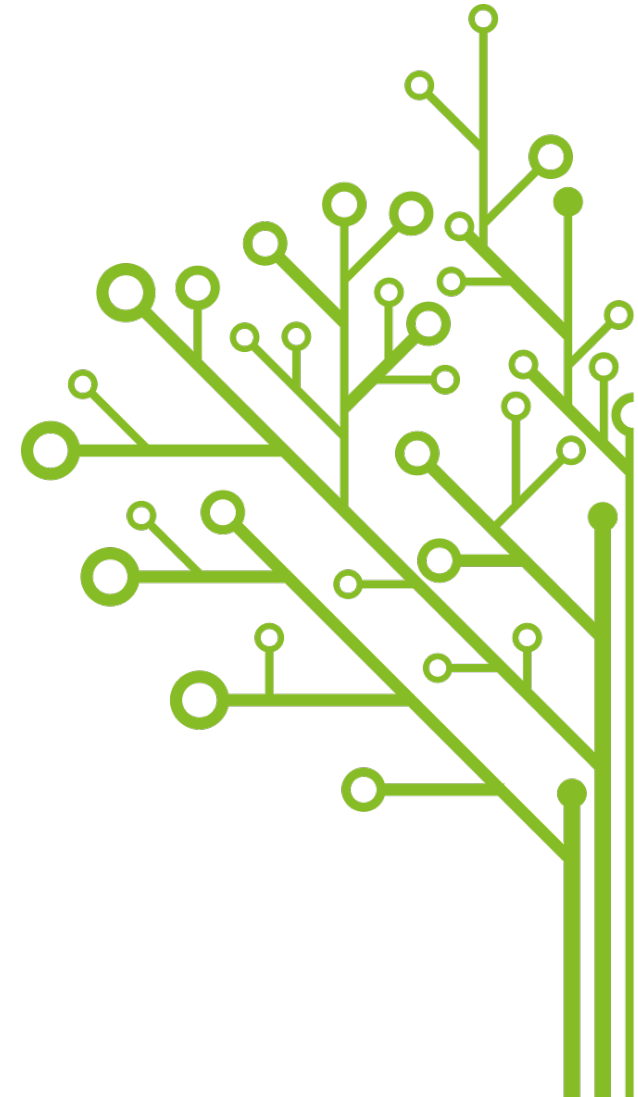
Vaihe 3: Mittaa ja vertaile

- Mittaa resurssien käyttöä (CPU, muisti, verkko, tallennustila).
- Vertaa versioita (ennen muutoksia vs. muutosten jälkeen).
- Seuraa kehitystä ajan yli.
- Priorisoi toistettavuus täydellisen tarkkuuden sijaan.

*Energiatehokkuus ilman
mittaamista on pelkkää arvailua.*

Yhteenvetona:

- Kestävyys on sisällytettävä koko elinkaareen.
- Vihreä koodi vähentää tarpeetonta ajonaikaista energiankulutusta.
- Aloita pienesti: tunnista → sovela → mittaa → kehitä iteratiivisesti.
- Energiatehokkuus on ohjelmiston laatuominaisuus.



Lähteitä

- Joof, M. B., Khan, M. A., Oyedeji, S., & Porras, J. (2026). Integrating API energy profiling into developer workflows: The VerdeFlow prototype. In Proceedings of the 10th International Workshop on Green and Sustainable Software (GREENS'26). ACM.
- Becker, C., Betz, S., Chitchyan, R., Duboc, L., Easterbrook, S.M., Penzenstadler, B., Seyff, N. and Venters, C.C. (2016) *Requirements: The key to sustainability*. IEEE Software, 33(1), pp.56–65.
- Hackenberg, D., Schöne, R., Ilsche, T., Molka, D., Schuchart, J. and Geyer, R. (2015) *An energy efficiency feature survey of the Intel Haswell processor*. IEEE International Parallel and Distributed Processing Symposium Workshops, pp.896–904.
- Hilty, L.M. and Aebischer, B. (2015) *ICT for sustainability: An emerging research field*. In: Hilty, L.M. and Aebischer, B. (eds.) ICT Innovations for Sustainability. Springer, pp.3–36.
- Penzenstadler, B., Bauer, V., Calero, C. and Franch, X. (2014) *Sustainability in software engineering: A systematic literature review*. Proceedings of the 16th International Conference on Evaluation & Assessment in Software Engineering (EASE), pp.1–10.
- Procaccianti, G., Lago, P. and Vetro', A. (2014) *Energy efficiency in software: A systematic literature review*. Journal of Systems and Software, 97, pp.135–155.
- Partanen, L., Sipilä, A., Hague, S., Porras, J. (2025). Mapping of the system of software-related emissions and shared responsibilities. 2025 IEEE/ACM 9th International Workshop on Green and Sustainable Software (GREENS), Ottawa, ON, Canada, 2025, pp. 39-46.
- Kivimäki, L., Partanen, L., Porras, J., Tarkkanen, K., Tuikka, A-M., Mäkelä, J-M., Mäkilä, T. (2024) Building up Green Software Life Cycle Model. In: 2024 10th International Conference on ICT for Sustainability (ICT4S), Stockholm, Sweden, 2024, pp. 20-28.

Kiitos.

Laura Partanen
LUT Software Engineering

laura.partanen@lut.fi



Elinvoimakeskus



Follow LUT Software
Engineering on social media