Green Software Engineering – Substance or fad?

Hans Christian Benestad
June 30, 2009
Ph.D. Trial Lecture
What does it mean to go green?

**Acid Rain**
Effects felt through the food chain

**Deforestation**
Deliberate by man

**Toxic Waste**
Man's poisonous byproducts

**Air Pollution**
Harming our atmosphere.

**To counteract the threats to our environment**

**Ozone Depletion**
Losing the Earth’s protective layer

[National Geographic, 2009]
Going green are by many regarded a hype

A hype or a product of a hype is not necessarily without substance
Hypes can be problematic, but are also powerful

Commercially exploited

Hampers nuanced debates

May end in disillusion

Global awareness

Change people’s behaviour
A definition of GSE, based on definitions of *Software Engineering* and *Going green*

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software...

+ 

To counteract the threats to the environment

An approach to developing, operating and maintaining software that recognizes and counteracts threats to the environment
How can SE counteract threats to the environment?

- Develop Toyota Prius power management software
- Software to predict climate changes
- Reduce CPU cycles generated by SW code
- Develop video conferencing
- Dispose less hardware
- Use video conferencing
- Reduce energy consumption in computing centers
- Websites for trading CO2 quotas
- Remove need for printouts and post
- Use OS with better power management
- Downloadable books
- Optimized air traffic control
- Cloud computing
- SW tools for distributed development
Can software engineers contribute to a greener planet?

Software-intensive technology plays a key role in counteracting threats to the environment

Should development of such technology be denoted Green Software Engineering?

“An approach to developing, operating and maintaining software that recognize and counteract threats to the environment”

Reduce the environmental impact of the software products and processes in themselves
Green software engineering is one enabler for *Green IT*

The ICT industry emits 2% of global CO2 emissions, most resulting from the power consumption of PCs, servers and cooling systems. [Gartner Group 2007]

- improve energy efficiency
- lower greenhouse gas emissions
- use less harmful materials
- encouraging reuse and recycling

[Murugesan, IEEE IT Professional 2008]

What can software engineers do?
Energy efficiency crosscuts the system architecture and the software engineering process

<table>
<thead>
<tr>
<th>Requirement Engineering</th>
<th>Establish requirements to energy efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Design &amp; Software Architecture</td>
<td>Use platform with appropriate power management capabilities</td>
</tr>
<tr>
<td></td>
<td>Choose environment supporting energy profiling</td>
</tr>
<tr>
<td></td>
<td>Design an event-driven architecture</td>
</tr>
<tr>
<td>Implementation</td>
<td>Energy-efficient programming</td>
</tr>
<tr>
<td></td>
<td>Implement power management interfaces</td>
</tr>
<tr>
<td></td>
<td>Do energy profiling to find hotspots</td>
</tr>
<tr>
<td>Test</td>
<td>Integrate energy profiling in test procedures</td>
</tr>
<tr>
<td></td>
<td>Test suspend/resume functionality</td>
</tr>
<tr>
<td>Improvement Loop (Feedback)</td>
<td>Establish field monitoring procedures</td>
</tr>
</tbody>
</table>

[GSE, Gerald Kaefer, Siemens AG, presented at ICSE 2009]
Coding style is not always irrelevant to energy efficiency

13 http requests to handle pictures

Assume 100 million views per day, and conservative estimates of other relevant parameters

Annual CO2 emissions of 500 kg

A simple change in the code would eliminate the emission

<FilesMatch "\.(png|gif)$"> Header set Expires "Thu, 15 Apr 2010 20:00:00 GMT" </FilesMatch>
Creating software that fits in a *cloud computing* model

**Today’s computing**

Scaled for max usage
Available resources are typically idle

**Cloud computing**

Dynamically scalable infrastructure services
Overall, unbeaten resource utilization
In design of embedded systems, energy consumption is a critical factor.

Thermal design

Power supply design

Weight

Cost

Operating time

Models to predict and optimize energy consumption are well established

Adapt to IT systems?
Engineering more eco-friendly business processes

Challenge any need for printing, including pdf generation

Limit the need for transportation of humans and objects

Two examples Topic for more research?
Resolving and trading off conflicting goals are core challenges of software engineering.

- Cost vs. functionality
- Efficiency vs. usability
- Going green vs. *
- Deployment architecture vs. evolvability

Going green can be viewed as a new generic goal for SE, possibly in conflict with other goals.
Integration of green qualities into existing SE frameworks?

Product quality models
- Portability
- Maintainability
- Usability
- Reliability
- Efficiency
- Functionality
- Adaptability
- Installability
- Co-existence
- Replaceability
- Portability compliance
- Analyzability
- Changeability
- Stability
testability
- Maintainability compliance
- Understandability
- Learnability
- Operability
- Attractiveness
- Usability compliance
- Maturity
- Fault tolerance
- Recoverability
- Usability compliance
- Time behavior
- Resource utilization
- Efficiency compliance
- Suitability
- Accuracy
- Interoperability
- Security
- Functionality compliance

Process quality models
- Process management
- Project management
- Engineering
- Support

Meta quality models
- ISO 9126
- CMMI
- S3M
- ISO 12207

SE Methods and practices
- Extreme Programming Explained
- Best Practices
Integrating eco-friendliness into product quality models

External and internal quality

- Functionality
  - suitability
  - accuracy
  - Interoperability
  - Security
  - Effect on environment
- Portability
  - adaptability
  - installability
  - co-existence
  - replaceability
- Maintainability
  - analysability
  - changeability
  - stability
  - testability
- Usability
  - understandability
  - learnability
  - operability
  - attractiveness
- Reliability
  - maturity
  - fault tolerance
  - recoverability
- Efficiency
  - time behaviour
  - resource utilisation

Eco-friendliness
- Energy consumption
- Hardware compatibility
- Document generation
- Transportation needs

Environmental effect part of functional design

Environmental side-effects of the product
Integrating eco-friendliness into software process models

CMMI process areas

- Process management
  - process definition
  - training
  - Innovation
  - Eco-consciousness

- Project management
  - planning supplier agreement
  - risk management
  - Eco management

- Engineering
  - requirements technical solution
  - verification
  - validation

- Support
  - vonfig management
  - QA
  - measurement

"Agile values"

- Individuals and interactions
- Working software
- Customer collaboration
- Responding to change
- Following a plan

Going green

Neglecting the environment

Or development methods

Working more environmentally friendly

17
Two prospects for the future
Building a giant cloud computing centre in the Norwegian mountains*

- Rich access to cool and secure mountain halls
- Stable political climate and power supply
- 99% of our electricity is hydroelectricity
- Financial strength

Would this make the cloud an attractive option for the largest IT users (public sector, finance) worldwide?

* [IKT-Norge/Teknisk Ukeblad]
Improved global navigation satellite systems will provide dependable and accurate positioning services.

Right human or object at the right time and place.

Small and inexpensive receivers.

Opens a huge market for positioning-based services, most of them software intensive.

New challenges for software engineering?
A new sub-profession for software engineers?
Conclusions

Software engineers have immense opportunities for counteracting environmental threats

Green software engineering is best understood as elements within existing product and process frameworks

...and research opportunities

- A better taxonomy for green elements in SE
- A framework for analyzing costs and benefits of green elements in SE
- Energy consumption models for IT systems
- Industry survey of current and future acceptance of green elements in software engineering