Literature Review

Green Grid Data Center Power Efficiency Metrics: PUE AND DCIE


This report is produced by a consortium of industries including the Hardware and Software Manufacturers who propagate the Cleaner and Green IT concepts. Certain metrics are derived by this group to measure the Power Usage Effectiveness (PUE) and Data Center Efficiency (DCE) which will allow a particular data center to estimate the efficiency and compare the same with other centers to take measure to improve the same.

The DCE is the reciprocal of PUE, and an improvement is made and the infrastructure aspect is added to DCE and hence is called DCiE, Data Center Infrastructure Efficiency.

The Green Grid develops metrics to measure data center productivity as well as efficiency metrics for all major power-consuming subsystems in the datacenter.

Southern California Edison Smart Grid Strategy & Roadmap

2007

This paper deals with the Smart Grid and the need and vision to achieve the goals of the sustainability.

Vision of a smart grid is to develop and deploy a more reliable, secure economic, efficient, safe and environmentally-friendly electric system. This vision covers all facets of energy from its production to transmission, distribution, and finally its efficient use in homes, businesses and vehicles. This smart grid will incorporate high-tech digital devices throughout the transmission, substation and distribution systems and integrate advanced intelligence to provide the information necessary to both optimize electric service and empower customers to make informed energy decisions.

With these statements from the policy makers the imperative conclusion drawn is that the systems need IT systems to develop and manage the smart grid. The Smart Grid development methodology is based on the following principles.

Customer Focused System Engineering

Open Innovation

Technology Development Scenario Planning

Proactive Standards development
Rigorous Technology Evaluation

Of all the above themes the most complex theme is that of the Information and Connectivity

*Futures of global interdependence (FUGI) global modeling system Integrated global model for sustainable development* 2004 Onishi A

The FUGI (futures of global interdependence) global modelling system has been developed as a Scientific policy simulation tool of providing global information to the human society and finding out possibilities of policy coordination among countries in order to achieve sustainable development of the global economy under the constraints of rapidly changing global environment.

The scientific integrated economics design concept of FUGI global modeling system has been influenced by the recent advancement of life science, biotechnology and information technology. The keywords are given below. (1) Lifeinformatic economics coupled with life science, information technology and economics. (2) Global dynamic cooperation and policy coordination among the countries, (3) Self-organization in accordance with changing environment, (4) automatic error correction system, (5) Brain physiology economics in Collaboration with right and left brain, (6) Fluctuation phenomenon considering alternative composite policy scenario projections under uncertainty world and (7) Global early warning system for geographical and global risks. All the above aspects enunciated indicate that the factors are common across the world and the evolution of a common goal of global sustainability of the entire ecosystem where business and administration is conducted for the well being of countries and continents.

This ever pervading phenomenon has influenced the ever advancing communities to take lead in preserving the environment and sustain the ecosystem against the indiscrete usage of the natural resources available to us in the nature.

*GITAM: A Model for the Adoption of Green IT*

*Molla A Dec 2008,

Green IT Adoption Model (GITAM) from a process technology perspective, green process technologies can be classified as end of pipe technologies and clean
End of pipe technologies reduce the environmental impact of emissions without necessarily changing the production process. On the other hand, clean technologies cause significant changes in the production process and their adoption is intended to reduce the level of environmental impact along a product’s life cycle from design to consumption. From a supply chain perspective green supply chain refers to integrating environmental thinking into the product design, sourcing, manufacturing, warehousing, distributing and end of life product management aspects of a supply chain.

The different perspectives are utilized the address the landscape of the Green IT Initiatives

- From an operation perspective,
- From a sourcing perspective,
- From a service perspective,
- From end of IT life management perspective

While the perspectives are being driven by several factors like socio economic and technology factors these are dealt in detail how these aspects affect the industry.

**Value models for engineering of complex sustainable systems**

*Arnautovica E and Svetinovic D 2010*

Sustainable Systems are complex systems-of-systems that pervade across several technical domains and organizations. They could include disparate sub-systems such as energy systems, information technology systems, transportation systems, buildings, etc. The author proposes to integrate Value Network Models (VNMs) into analysis and design of complex sustainable systems. VNMs include, e.g., actors, value objects, activities that generate value, and their dependencies. This paper considers the socio-technological factors that cover the entire spectrum of the sustainability factors, in a sense it is viewed as the factor that exchanges the value in an networked environment that make use of the systems and subsystems to effect this factor. In effect these networks gain certain economic value and are called value networks. These are then made use of to evaluate the costs that are incurred to build and manage the systems envisaged. These systems are called E-Value networks with the composition of Actors, Market Segment, Value Activity Value Object, Value Port, Value Interface, Value Exchange, and Dependency Path

Conclusively the Model arrives at the cost of the engineering in the engineering system itself and helps arrive at cost factors for sustainable engineering.
Green Software and Green IT: An End User Perspective
Kern E, Dick M, Johann T, Naumann S 2011

This paper deals with the End User perspective and that it considers the following aspects

Testing of Software Selection and Configuration, Results and End User Advices

The basic motivation in this paper is drawn from the Corporate Social Responsibility World Economics, Global Warming and Climate Changes, Sustainable ICT, Power Consumptions across the worlds for ICT systems and Movements in Green IT. The three aspects treated in this paper are the hardware, software and information availability to the end user. If we consider the overall landscape of the IT the operating systems choices are limited while the applications that are run on those operating systems will be many. Type of users also make a difference on the power consumed for running different application typical to them in case of Professional and home users.

This led to the test scenarios which finally concluded that the normal end user had consumed more power than the professional user in a typical scenario of browsing YouTube and Google maps. This leads to the belief that the general user applications have more power requirements that would increase with time.


This paper introduces the concepts that will help us to set Sustainable performance metrics for development of software while considering the economic, environmental and social aspects of software product development.

Sustainability properties can be summarized as below with above goals being addressed.

Development Parameters addressed are

- Reusability
- Portability
- Supportability

Usage parameters

- Performance
- Dependability
• Usability
• Accessibility

Process Parameters
• Predictability
• Efficiency
• Project Foot Print

With the above considered parameters we would develop a clear model for the sustainability measurement in the space

*Impacts of Information and Communication Technologies on Environmental Sustainability: speculations and evidence May 2001*

*Berkhout F & Hertin J*

This Paper deals with different order effects demarcated by the author of the ICT on Sustainability.

The first order effects, the negative ones that stem from direct use of the hardware and their production and disposal in terms of hardware manufacturing. While the positive effects can be mentioned as the effects that help environment in terms of electronic monitoring of toxic emissions, remote sensing for weather monitoring in terms of predicting natural disasters

Second Order Effects: These can be said as those effects that effect the business growth due to infusion of the ICT into the business.

Efficiencies are attributed to the growth of the ecosystem in this case for e.g.

- Intelligent Automated Production Processes
- Intelligent Design and Usage of Products
- Re organization of Supply chains and Business re alignment
- Intelligent Logistics and Distribution
- High end networking for efficient management

Third Order Effects:

These effects are still to be enumerated in terms of concrete measurable growth or negative growth parameters in terms of GDP growth of the nation.
Towards Software Sustainability Guidelines for Long-living Industrial Systems

Sustainable Architecture: Several heterogeneous approaches are known for designing Sustainable software architectures. Scenario-based / usage of the software based architecture evaluation methods, can be used to elicit evolution scenarios from various stakeholders and to discuss their impacts on software components and reveal potential ripple effects through the architecture.

Change-oriented architectural style adopted by software product lines help to manage system variability, while architecture-centric model-driven development is a potential mean to cope with platform adaption. Risks associated to sustainable software architecture design are for example undocumented design rationale or too much flexibility. Non documentation of the design rationale complicates the development and maintenance of the software thus leading to the degradation of the life of the software and leads to non reusability. While too little flexibility in the architecture limits scalability options this inducing rigidity the software ecosystem while too much flexibility increases architectural complexity and thus maintainability (i.e., understandability, analyzability, modifiability).

GREEN IT: The Global Benchmark
A Report on Sustainable IT in the USA, UK, AUSTRALIA and INDIA
Porritt J, 2010

This paper examines the maturity of Green IT in large IT-using organizations in the United States of America, the United Kingdom, Australia and India.

- Green IT Lifecycle (Procurement and Disposal)
- End User IT Efficiencies
- Enterprise and Data Center IT Efficiencies
- Usage of IT as a Low-Carbon Enabler
- Green IT Measurement and Monitoring

The Framework by the author would guide as a starting principle for embarking on an endeavour that comprehensively addresses the Green IT initiative from an industrial perspective.
Fostering Green Strategy Through “A-F-F-I-R-M” Model For Sustainable Development within the Environmental Socio-Economic System in Malaysia

Abu Bakar K, Rajiani I, Razali Ayob Md, Shwal Nasri N, 2011

“A-F-F-I-R-M” model is prorogated by the Malaysian Government and is supported by the industry to complete ecosystem of the environment sustainability.

Energy is the catalyst for development. Globally, the per capita consumption of energy is often used as a barometer to measure the economic development level of a country. Realizing its importance as a vital component in economic and social development, the Malaysian government has continuously reviewed its energy policy to ensure long-term reliability and security of energy supply.

Green Technology is identified as a driver for the economic growth of the state of Malaysia. This in turns results in identification of the four pillars of focus for Green IT initiative.

- Energy - Seek to attain energy independence and promote efficient utilization;
- Environment – Conserve and minimize the impact on the environment;
- Economy – Enhance the national economic development through the use of technology; and
- Social – Improve the quality of life for all.

These are supported by the following deployment drivers namely:

- Awareness
- Faculty
- Finance
- Infrastructure
- R&D and Commercialization
- Marketing

The basic tenets of any enterprise model for Green IT Initiative are based on the guidelines given by the Government.

Software or Hardware: The Future of Green Enterprise Computing

In the enterprise, a system’s energy consumption has become a significant portion of its total lifetime cost. In order to evaluate energy conservation techniques for enterprise computing in a crisp, quantitative way, author takes an economic view and consider a purely cost standpoint, rather than consider moral or social motivations.

This paper typically considers the computing systems as any other system that consumes power for its sustenance and hence the same principles applied elsewhere for the energy conservation are applied. Consequently the Desktop systems could be replaced with the Laptop or docked stations that consume lesser power and lesser space.

Also the migration from the existing systems to laptop systems hardly saves significant amounts considering the software architectures deployment to manage such systems is far more cost incurring than optimizing. Then the focus shifts to the power management systems that are deployed for the management of the current systems. This leads us to the study of the centralized energy management systems and Grid computing and energy systems.

A Method for Alignment Evaluation of Product Strategies among Stakeholders (MASS) in Software Intensive Product Development

Dick M, Naumann S 2010

This paper says that the emergence of markets for off-the-shelf/packaged and embedded software market-driven development of software and software intensive products is gaining increased interest/attention compared to customer-specific system development. Consequently, a shift in focus is occurring, affecting software development in general and requirements engineering and product management in particular. Consequently, product management is faced with several challenges that have to be addressed as a part of the market-driven requirements engineering process. One of the important challenges is how to select the right mix of requirements, balancing short-term and long-term gains. One way to address this challenge is to utilize product strategies for selecting requirements. However, in order to do this the internal success-critical stakeholders (SCS) involved in strategies creation and requirements selection need to be aligned with respect to a product’s strategic goals and objectives. This paper presents a method to enable the evaluation of degree of alignment between SCS with respect to the understanding and interpretation of a product’s strategy. Further, the method not only enables the evaluation of alignment, but also specifically shows misalignment, and enables the identification of leading causes. The model/method have evolved in collaboration with industry.
In principle we have two views in this model:

- the product view, which is given by the lifecycle model
- the organizational view.

The models can be best described as below

- a lifecycle model for software products
- a procedure model and
- recommendations for actions and tools

This aligns the developing organization to consider these criteria during development and to apply further activities in order to assess impacts that result from the software product over its whole life cycle.

*Enhancing Software Engineering Processes towards Sustainable Software Product Design. Dick M, Naumann S 2010*

It is well known that global warming, greenhouse gas effects, climate change and sustainable development are key challenges of the current day businesses. Information and Communication Technology (ICT) takes an important part within these challenges. Though ICT can optimize material flows and therefore reduces energy consumption would also consume more and more energy. Several studies discuss the impact of ICT on the environment or consider the balance between energy savings and energy consumptions by ICT. Especially, to date it is not clear, whether energy consumption by ICT is greater or smaller than energy savings by ICT, e.g. because of more efficient processes or simulations of scenarios. There has been no concerted effort to collect and collated data that can deduce and establish the facts if ICT is consuming more energy or not. Consequently, considering problems like climate change, the reduction of energy and resource consumption, which is caused by software, is necessary. Therefore approaches and solutions for the development and usage of sustainable software are essential.

However, famous software engineering methodologies do not cover issues about how to integrate environmental or sustainability aspects into software design and development.

There are several models that are proposed in this paper that enunciates the process to develop sustainable softwares in the industry. We need to look at the applicability of such models in the environment where these models need to be deployed.

*Integrating the Complexity of Sustainability in Requirements Engineering Mahaux M, Canon C 2010*
This paper addresses the inevitability of the process of adoption of sustainability concepts into the development of Software holistically and that this new field will have no other option than integrating this complexity into its design practices through opening collaborations with sustainability experts. This in turn will result in making requirements management a very complex process where human behaviours affect the requirements indirectly while building a system.

There are several level impact on the sustainability in the development of software while the First level impact is resource use and pollution from mining, hardware embedded software production, power consumption, and disposal of electronic equipment waste. Second level impact would be result from indirect means of using ICT, like energy and resource conservation by process optimization (dematerialization effects), or resource conservation by substitution of material products with their substitutes. Third level impacts are long term indirect effects that result from ICT usage, like changing life styles that promote faster economic growth that fuels the reversal of such measures being thrown out of gear resulting in faster rate of acceleration of carbon emissions.

*Life-cycle Assessment in Software Engineering; Galster M 2010*

This paper not only aims at treating sustainability as primary product requirement, but also considers the process of developing the software product.

Life-cycle assessment, as applied to conventional product development, can be applied in the Software Engineering domain to determine the environmental impact of a software product itself, but also the environmental impact of a software development process.

Software engineering if looked at as a system engineering angle it would direct us to the view of Life cycle management of the Product in the context of software being a product with all other components contributing to its development in terms of hardware and power systems and other resources.

*Profiling Energy Usage for Efficient Consumption*

*Chheda R, Shookowsky D, Stefanovich S, and Toscano J 2010*

This paper provides an insight into details of the mathematical calculations of the Power consumption and carbon emissions. There are several examples given and explained on how the same can be extended to the overall ecosystem of the Green IT initiative.
A typical symptomatic treatment is given to the entire life cycle of a software program development to the usage.

Different types of software programs are used for such profiling such as Web, Databases and Applications. These are profiled against the CPU processing power consumed while executing such software programs and the same is documented.

Similar profiling is done against the operating system used to run such programs. The philosophy is extended to the networks deployment.

*Architecting for Sustainable Software Delivery*

*Koontz R J, Nord R L 2012*

This paper deals with the top architectural practices that military systems follow for a sustainable software delivery. While the battlefield systems are ever evolving in size and requirements that need enormous amounts of power to run while the budgets allocated are diminishing.

The business case for an over extended system from an existing system to meet the end user objectives without causing entropy in the system is far from reaching the intended goal.

Over a product lifecycle, business goals and objectives continue to evolve as capabilities are realized. This article describes the role that five architectural practices are continuing to play in enabling the stakeholder to achieve long-term business goals. Agility, while balancing flexibility and stability enables architectural development to follow a “just-in-time” model that complements iterative and incremental enhancement development and integration. Delivery of capabilities is not delayed pending the completion of exhaustive requirements and design activities and reviews. Incremental Iterative Development and informed technology insertion decision making, Frequent Architectural maintenance, strategies and patterns usage in architectures will make the end goals of sustainable delivery a reality.

*Integrating hardware, software and mindware for sustainable ecosystem development: Principles and methods of ecological engineering in China*

*Wang R , Yan J 1998*

The paper deals with the key integration of ‘hardware’, ‘software’ and ‘mindware’. Eight design principles of ecological engineering based on eco-cybernetics are
discussed, which fall into three categories: competition, symbiosis and self-reliance.

The fundamental tasks of ecological engineering are to develop a sustainable ecosystem through the integrative planning of its structure, function and processes by encouraging totally functioning technology, systematically responsible institutions and ecologically vivid culture.

Ecological engineering: an integration of hardware, software and mindware.

*Sustainable Approaches and Good Practices in Green Software Engineering*

*Shalabh Agarwal, Asoke Nath, and Dipayan Chowdhury February 2012*

This paper presents the present day scenario of the ethical obligations of the industry while the technical advancement obligates the usage of energy for such advancement. This also on a longer term perspective is a profitable proposition for the entire industry. The authors focus on the facts of pollution of environment and the net effect on the ecosystem. This is linked to the need to tweak the Software Development Life Cycle that contributes enormously to increase in the carbon footprint every day.

Unfortunately the end user is not involved in the choice of software that he needs as not very many systems with good stability are available for him. Hence non energy efficient systems are deployed resulting in high energy consumption; hence the software side also influences the carbon footprint of the IT.


The author of this paper proposes a Greensoft Model that paves way for Software oriented engineering process for achieving sustainability in software industry. The resource and power consumption of ICT is still increasing, but also the benefits of ICT, e.g. in finding more efficient solutions for environmental problems. However, there is still a lack of models, descriptions or realizations in the area of computer software and software process models. Author proposes the basic definitions of the terms “Green and Sustainable Software” and “Green and Sustainable Software Engineering”, then outline a conceptual reference model, the GREENSOFT Model. This model includes a cradle-to-grave product life cycle model for software
products, sustainability metrics and criteria for software, software engineering extensions for sustainably sound software design and development, as well as appropriate guidance.

*Measuring Software Sustainability, OSS Watch, Lee S 2009.*

To build software that can be sustained over a period of time needs to be both useful and adaptable as the users' needs evolve. There should be a consideration for sustainability when architecture is being designed, for potential for reuse and is therefore a key factor in the sustainability of software.

Reuse can save time and money, and increase the reliability of resulting products while at the same time throws a paradox that its technical obsolescence may lead to non usability with the impending technology advancement.

However, an attempt to reuse software that is not easily reusable can have the reverse effect software stability and hence the sustainability factor is defeated. As this is also affected by the number of environments in which the software is likely to be used and reused. This factor has an implication on the design of such software architectures as it makes a significant impact on the communities that utilize this software which are dispersed across different domains of applications.

Structural integrity, understandability, Completeness, Conciseness, Portability, Consistency, Maintainability are the foundation pillars of the sustainability measurement while different ways of attributing values for measure of performance will need to be evolved based on the domain or area that the software to be built is addressing.

*Energy-synchronized computing for sustainable sensor networks*

*Zhu T, Zhong Z, He T, Zhang Z-L 2010*

This paper author deals with the sustainable operations of wireless sensor systems, and environmental energy harvesting.

It has been regarded as one of the most fundamental solutions for long-term applications that the energy to run such systems be drawn from the environment in such a way that it sustains for the period intended.

In energy-dynamic environments, energy conservation is no longer considered necessarily beneficial, because energy storage units (e.g., batteries or capacitors) are limited in capacity and leakage-prone. In contrast to legacy energy
conservation approaches, we aim at energy-synchronized computing for wireless sensor devices. Design of such systems will become the starting point to take forward the concepts of environment drawn energy systems. To efficiently use the harvested energy, the system is designed and implements leakage-aware feedback control techniques to match the activities of sensor nodes with dynamic energy supply from environments. Results indicate our leakage-aware energy-synchronized control can effectively utilize energy that could otherwise leak away.

Researchers have designed various types of energy harvesting technologies to collect ambient energy from the environments. These include solar, wind, kinetic, piezoelectric strain, and vibrational energy. Various solutions have been proposed for energy efficiency at various levels of the system architecture, ranging from energy-efficient hardware, LPL link layer, topology management, node placement, sensor clustering, sensing coverage, data dissemination, data aggregation, in-network caching and storage, up to application-level energy-aware designs.

Sustainability Guidelines for Long-Living Software Systems

Zoya Durdik_, Benjamin Klatt_, Heiko Kozioleky, Klaus Krogmann_, Johannes Stammel_ and Roland Weiss 2012

This addresses the factors that influence the Sustainable software development while building economically sustainable software systems in response to the business landscape changes with the passage of the time. Sustainability infusing techniques and how these techniques are deployed with the lessons learnt during the past two decades is dealt in this paper with an overview of such lessons learnt. A catalog of “software sustainability guidelines” to support project managers, software architects, and developers during system design, development, operation, and maintenance is presented along with how these are derived these guidelines and The guidelines are ordered as Requirements, Architecture, Design, Implementation, Validation and Verification, and Maintenance.

The lessons learnt are made practical by adopting them in a manner how these are applied, that will provide the directions of deployment of such lessons learnt as follows.

Incentives may be reinforced through activities, such as an explicit documentation of the quality and sustainability goals, enforced by a regular review process.

Evolutionary architecture practice approach makes it more probable the development of sustainable software while retaining the business goals. This proposes incremental and iterative refinement of software architecture in context of unknown or rapidly evolving business and technical requirements.