|  |  |  |
| --- | --- | --- |
| **Work Title** | **Dimension/Category** | **Issue and Metrics** |
| Evaluating Code Sustainability: A Comprehensive Study of Metrics and Tools | Environmental- code sustainability | %CPU usage, %memory usage, #code smells |
| GS3M Model- A conceptual model- attributes, sub attributes | Environmental, Economic, Technical, Social, Individual | **Environmental**: **Energy Consumption**- energy efficiency, runtime efficiency, CPU-intensity, memory usage, peripheral intensity, idleness and algorithmic efficiency  **Resource Consumption**- amount of hardware that needs to be replaced before the sustainably worthwhile lifetime is reached  **Technical**: **Perdurability**- Functional evolution, Technical evolution  **Social: software social added values**- Accessibility  **Economic**: **Software process evolving intellectual capital-** Customer capital value, Human capital value, Structural capital value  **Low cost software process**- Market requirements value, Physical value w.r.t. Cost  **Long term profitable software**- Innovation value for market, Differential value |
| Towards Incorporating Sustainability while Taking Software Product Management Decisions | Human, Social, Economic, Environmental | **Social**: Customer capital value, network externalities  **Economic**: Maintainability value, innovation value, differential value, physical value  **Environmental:** Market requirements value, Physical value, sustainability value of technology, product intrinsic value |
| The GREENSOFT Model: A reference model for green and sustainable software and its engineering | Environmental | **Common quality attributes for sustainability**- Portability, Usability & Accessibility, Efficiency, Predictability, Modifiability & Reusability, Project's Footprint  **First order criteria**- Energy Efficiency, Energy Consumption, Hardware Requirements, Resource Consumption (during development), Resource Consumption (during use)  **Indirect criteria**- Social effects, rebound effects |
| Incorporating Sustainability Design in Requirements Engineering Process: A Preliminary Study | Economic, Social, Technical, Environmental  Case study of-  Climate Monitoring System | **Economic**- Cost efficiency, minimize development cost  **Social**- Better climate understanding  **Technical**- Fast performance, usability, scalability, accuracy  **Environmental**- Energy Efficiency |
| Risks and Requirements in Sustainable App  Development—A Review | Social, Environmental, Economic | **Social:** Ease to use, adapt, modify, maintenance through patches, learnability, user error protection, replacement ability, social engineering  **Environmental**- Longitivity, energy efficient maintenance, energy efficient data management  **Economic**- long term profit, compatibility, customer oriented management |
| **Green Software Quality: A Comprehensive Framework for Sustainable Metrics in Software Development** | Environmental, Technical | **Environmental: Energy Efficiency-** Energy consumption per transaction, power usage effectiveness, algorithmic complexity  **Resource Efficiency-** CPU utilization, memory usage, storage efficienc**y**  **Technical:** Maintainability index, code complexity, software longitivity |
| **A Taxonomy and Future Directions for Sustainable Cloud Computing: 360 Degree View** | Environmental, Technical, Economic | **Environmental-**  Carbon Usage Efficiency (CUE),  Water Usage Efficiency (WUE), Energy Reuse Effectiveness, Green Energy Coefficient, The Green Index, Energy Proportionality, Energy Consumption, Energy-Efficiency, Average Datacenter Efficiency, Computation Power Consumption, Power Usage Effectiveness (PUE)  **Economic-** Total Cost of Ownership, Return on Investment, Capital Expenditure, Capacity, Memory Usage, Storage Usage  **Technical-** Execution Time, Energy Cost or Energy Expense, Network Bandwidth, VM Co-location Cost, Resource Utilization, Network Power Usage, Latency, Storage Throughput |
| **Software metrics for green parallel computing of big data systems** | Environmental, Technical | **Technical-** Maintainability index (MI), Cyclomatic complexity, Technical debt, Software reusability, Modularity, Scalability  **Environmental-** Power consumption, CPU utilization, Memory utilization, Disk utilization, Network utilization,  Code complexity |
| **A Systematic Review for Sustainable Software Development Practice and Paradigm** | Technical | maintainability, reusability, coupling, cohesion, Module Size Boundedness Index, Quality of Modularization, Layer Organization Index, Cyclic Dependency Index, Density of architectural scent |
| **Sustainable Approaches and Good Practices in Green Software Engineering** | Technical | Memory requirements, processor time, network bandwidth |
| **Green Measurements for Software Product Based on Sustainability Dimensions** | Environmental, Economic | **Environemntal:** Energy Efficiency:  electricity, power supply, consumed material, and emissions.  Resource Efficiency: CPU usage, memory usage, storage usage, and I/O usage  **Economic-** productivity |
| **Measuring Software Sustainability from a Process Centric Perspective** | Organizational Or Social | Conference call rate, meetings rate, compression ratio (effort/duration), team stability ratio, scope creep ratio, employee satisfaction campaigns |
| **The Green Software Measurement Structure Based on Sustainability Perspective** | Economic, Environmental, Social | **Economic-** usability, cost reduction, productivity  **Environmental: Energy Efficiency-** energy consumption, CO2 emissions.  **Resource Efficiency-** CPU usage, memory and storage usage, input/output usage rate  **Social-** Tool support, employee support |
| **Green Software Process Based on Sustainability, Waste and Evaluation Theory Approach: The Conceptual Model** | Resource, people, organization, technical, environmental, technology |  |
| **SustainScrum: integrating sustainability assessment in a tailored Scrum process for computing quantitative sustainability indicators** |  | Sustainability KPI reports |
| **Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow** | Social, Individual, Environmental, Economic, Technical | **Social**- Sense of community, Trust, Inclusiveness and diversity, Equity, Participation and communication  **Individual-** Health, Lifelong learning, Privacy, Safety, Agency  **Economic-** Value, Customer relationship management (CRM), Supply chain, Governance and processes  **Technical-** Maintainability, Usability, Extensibility and adaptability, Security, Scalability |
| **Do we really know what we are building? Raising awareness of potential Sustainability Effects of Software Systems in Requirements Engineering** | Social, Individual, Environmental, Economic, Technical | **Social**- Sense of community, Trust, Inclusiveness and diversity, Equality, Participation and communication  **Individual-** Health, Lifelong learning, Privacy, Safety, Agency  **Economic-** Value, Customer relationship management (CRM), Supply chain, Governance and processes  **Technical-** Maintainability, Usability, Extensibility and adaptability, Security, Scalability |
| **Enhancing Sustainable IoT Systems Through a Goal-Oriented Requirements Analysis Framework** | Technical | CPU usage, system latency, memory usage, code generation error rate, developer effort (lines of Code-LOC) |