Chapter 8
Sustainable Software Solutions for Business Project Management Integrating Eco-Conscious Practices:
Real-Time Applications in Healthcare Systems

Wajeed Mohammed Abdul
https://orcid.org/0000-0002-5047-7857
Lords Institute of Engineering and Technology, India

ABSTRACT

Green Software Engineering integrates computing practices with sustainability goals, energy efficiency, carbon-aware computing, and ecological responsibility in software development. The chapter deals with a) evolution, significance of GSE in the digital age. b)theoretical foundations, sustainable software metrics, and the impact of the software lifecycle on GSE. c) strategies and practices in GSE, including carbon-aware and energy-efficient programming, green DevOps, and eco-centric agile project management. d) Integrating GSE into business project management, environmental key performance indicators, and tools and frameworks for green decision-making. e)examines the challenges of implementing GSE. f)Case study of healthcare in GSE. g)discussion on systematic thinking in sustainable software. h) insights from theory to practice in GSE. i)overview of active research projects

DOI: 10.4018/979-8-3373-4652-6.ch008

in GSE at various universities. j) discussion on the future of GSE. This chapter is comprehensive and accessible to all readers, from beginners to research scholars interested in exploring GSE.

INTRODUCTION TO GREEN SOFTWARE ENGINEERING

Green Software Engineering (GSE) signifies a fundamental change in the software development process by incorporating environmental sustainability into essential computing activities. It is described as "a systematic approach to designing, developing, deploying, and maintaining software with minimal environmental impact, especially in terms of energy consumption and carbon emissions," (Rashid et al., 2021). GSE's scope extends beyond merely enhancing code efficiency or cutting down on energy use; it involves a comprehensive transformation of software practices to meet ecological and sustainability objectives throughout the software's lifecycle.

Historical Context and Evolution

The origins of Green Software Engineering can be traced to the early 2000s, coinciding with the rise of environmental informatics, a discipline dedicated to utilizing information systems for environmental oversight and management. Initially, environmental informatics focused on harnessing computing capabilities to tackle environmental challenges, such as climate system modeling, pollution monitoring, and ecological simulations, (Lu, Chang, & Liao, 2013). However, it did not thoroughly consider the environmental impact of computing itself, including energy consumption and the carbon footprint associated with data processing.

This awareness led to the concurrent development of Green IT, which broadened the environmental perspective to encompass IT infrastructure, emphasizing server efficiency, energy-conscious hardware setups, and sustainable data center operations, (Eshbayev et al., 2024). While Green IT concentrated on hardware and infrastructure efficiency, the software aspect remained largely unexplored. Over time, research began to reveal that software, even when executed on optimized hardware, could significantly contribute to energy inefficiency due to suboptimal algorithms, unnecessary computations, or inefficient execution paths, (Guo et al., 2021).

This shift in focus spurred the emergence of Green Software Engineering as a distinct subfield around 2010, redirecting attention from physical infrastructure to energy optimization at the code level, carbon-aware design principles, and environmentally intelligent development practices, (Kumar et al., 2024). The formalization of Green Software Engineering has been supported by frameworks like the Software