GREEN SCRUM MODEL: Implementation of Scrum in Green and Sustainable Software Engineering

R M D H LAVANYA

Dept. of Computer science, Rajamahendri Degree and P.G. College for women, Andhra Pradesh, India

---

Abstract - Sustainable development is the very much crucial term on which entire world is focusing on today. Agile methodology is one of the software practices that entire software companies revolving round it. Sustainable development is the idea that human societies must live and meet their needs without compromising the ability of future generations to meet their own needs. The Sustainable Development Goals are a universal call to action to end poverty, protect the planet and improve the lives and prospects of everyone, everywhere. Business organizations and ICT firms consider green computing as an integral part of their overall business strategies. From the business and user perspective, the main concern is all about development of software with no impact on climatic change and sustainable development. Literature reveals that in recent years a number of struggles have been made while creating green and sustainable software. Several researchers has focused on hardware oriented solutions for environmental sustainability whereas others have worked on software oriented solutions. Scrum is an agile framework for developing, delivering, and sustaining complex products, with an initial emphasis on software development. It is designed for teams of ten or fewer members, who break their work into goals that can be completed within time-boxed iterations, called *sprints*, no longer than one month and most commonly two weeks. The Scrum Team track progress in 15-minute time-boxed daily meetings, called daily scrums. At the end of the sprint, the team holds sprint review, to demonstrate the work done, and sprint retrospective to improve continuously.

***Key Words*: Green Software Engineering; Sustainability and Sustainability Dimensions, Software Product Life Cycle, software engineering, Scrum**

1. INTRODUCTION

There are two big terms which are ruling the present world: Sustainable development and Agile methodology. Sustainable development is the very much crucial term on which entire world is focusing on today. Agile methodology is one of the software practices that entire software companies revolving round it. Sustainable development is the idea that human societies must live and meet their needs without compromising the ability of future generations to meet their own needs. The "official" definition of sustainable development was developed for the first time in the Brundtland Report in 1987. Specifically, sustainable development is a way of organizing society so that it can exist in the long term. This means the preservation of the environment and natural resources or social and economic equity for the present and future generations. The Sustainable Development Goals are a universal call to action to end poverty, protect the planet and improve the lives and prospects of everyone, everywhere. The 17 Goals were adopted by all UN Member States in 2015, as part of the 2030 Agenda for Sustainable Development which set out a 15-year plan to achieve the Goals.

Today, At the core of the 2020-2030 decade is the need for action to tackle address the **climate emergency.** Business organizations and ICT firms consider green computing as an integral part of their overall business strategies [1].From the business and user perspective, the main concern is all about development of software with no impact on climatic change and sustainable development. Literature reveals that in recent years a number of struggles have been made while creating green and sustainable software. Several engineers has focused on hardware oriented solutions for environmental sustainability whereas others have worked on software oriented solutions [2]. Imtiaz and Mahmoud [3] have proposed green model having two levels, for sustainable software engineering, the first level presents guidelines for green software development and thesecond level is consist of methods occupied by the software during its execution to support green computing. GREENSOFT model for sustainable software has been presented by [4], the model supports software engineers / administrators / software users in building, sustaining, and using software in a more ecological way. Mahaux and Canon [5] have argued that requirements engineering is critical for sustainable software development in the whole software life cycle. The latest literature in the domain reveals that there is scarcity of software engineering models and tools that can well define how software can be designed and maintained in an eco-friendly manner [6]. Software engineering researchers have developed keen interest in green software development due to the demand of environmental sustainability [7]. Green Software has been defined in the literature as, that Software, whose direct and indirect negative influences on economy/humanity/ human beings/environment due to its creation, deployment, and usage minimal and/or which has a positive consequence on sustainable development [8]. Creating energy efficient and green software is becoming popular subject rapidly [9]. The different aspects of green and sustainable software engineering are summed up in the following definition: "Green and Sustainable Software Engineering is the art of developing green and sustainable software with a green and sustainable software engineering process". Summarizing these definitions, a sustainable software product ideally meets three conditions[4]:

- The software is produced in a way that meets sustainability objectives.

- The software has minimal negative social and environmental impacts during its usage (first-order effects).

- The software functionality reinforces sustainable development or at least has no negative impacts on the society or environment (second-order and systemic effects)

Agile software development methods have a number of reported benefits on productivity, project visibility, software quality and other areas. Agile software development (ASD) methods are often advertised as a contrast to the traditional, plan-driven approach to software development [9]. ASD methods are claimed to increase software quality [10], improve communication [11] as well as coordination [12] and increase productivity, to name just a few. The Agile Manifesto [11], created in 2001, lists a set of values upon which ASD relies. Along with these values, there is also a set of principles. Principles are "domain-specific guidelines for life" showing how the values can be applied in different areas. Thirdly there are practices, which are even more concrete, specific applications of the values and principles. Scrum is simple. It is the opposite of a big collection of interwoven mandatory components. Scrum is not a methodology. Scrum implements the scientific method of empiricism. Scrum replaces a programmed algorithmic approach with a heuristic one, with respect for people and self-organization to deal with unpredictability and solving complex problems. Sustainable or green software engineering is getting limelight among professionals and researchers [13].Mohhankumar and Anand Kumar [14] proposed green based model for sustainable software engineering. Recently Penzenstadler et al. [15] proposed a blueprint for a course on software engineering for sustainability. Scrum with sustainablility measures works very well for innovative and complex product development projects.

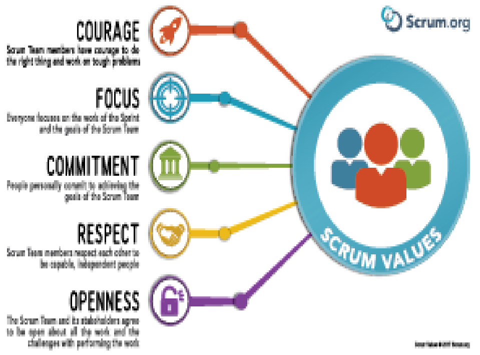
1.1 Scrum Frame work:

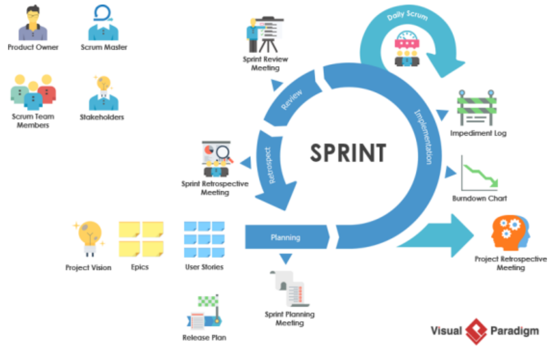
Scrum is an agile framework for developing, delivering, and sustaining complex products with an initial emphasis on software development that works very well for innovative and complex product development projects.

In July 2016, the Scrum Values were added to The Scrum Guide. These values include Courage, Focus, Commitment, Respect, and Openness.

The Scrum Team consists of a Product Owner, the Development Team, and a Scrum Master.

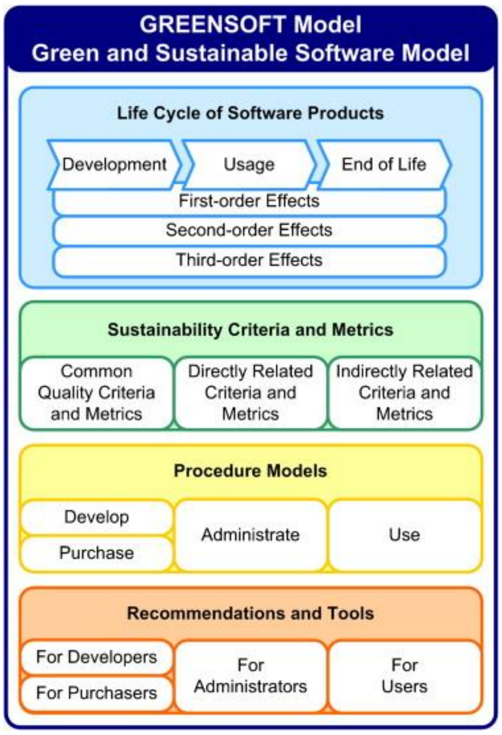
The product owner creates a product backlog (essentially, a wishlist of tasks that need to be prioritized in a project). The Scrum team conducts a sprint planning session where the tasks necessary to complete items on the wishlist is broken down into small, more easily manageable chunks. The team creates a sprint backlog and plans its implementation. The team decides a time duration for every sprint. (The most common intervals is probably two weeks). The team gets together every day for a brief Scrum meeting (often referred to as a Daily Standup) where each member of the team shares daily updates, helping the team and the project manager assess the progress of the project. The certified Scrum Master guides the team and keeps them focused and motivated. The stakeholders and the product owner conduct a review at the end of each sprint. This is the cycle followed by a Scrum team in a product development project. The three roles mentioned above - the Product Owner, the Scrum Team, and the ScrumMaster together play a major role in exercising this framework.


Scrum.org
COURAGE
Scrum Team members have courage to do the right thing and work on tough problems
FOCUS
Everyone focuses on the work of the Sprint and the goals of the Scrum Team
COMMITMENT
People personally commit to achieving the goals of the Scrum Team
RESPECT
Scrum Team members respect each other to be capable, independent people
OPENNESS
The Scrum Team and its stakeholders agree to be open about all the work and the challenges with performing the work
SCRUM VALUES



1.2 GREENSOFT Model:

The GREENSOFT Model (Fig. 1) classifies and sorts the described characteristics of sustainable software and its engineering [1]. This reference model contains four parts: the life cycle of software products; criteria and metrics that represent and measure sustainability aspects directly and indirectly related to the software product; procedure models for the different phases; and recommendations for action as well as tools.


GREENSOFT Model
Green and Sustainable Software Model
Life Cycle of Software Products
Development
Usage
End of Life
First-order Effects
Second-order Effects
Third-order Effects
Sustainability Criteria and Metrics
Common Quality Criteria and Metrics
Directly Related Criteria and Metrics
Indirectly Related Criteria and Metrics
Procedure Models
Develop
Purchase
Administrate
Use
Recommendations and Tools
For Developers
For Purchasers
For Administrators
For Users

2. PROPOSED MODEL:

Although much research has been made in improving energy efficiency of hardware and infrastructure, there is a gap of research in software engineering and how to design green software. Some of these researches are related to the practices we could adopt while designing, developing, testing and deploying software to not only reduce the energy consumption of resources but also to avoid negative impacts on sustainability.

There are two directions in Green software: GREEN BY SOFTWARE and GREEN IN SOFTWARE [HBR]. Nowadays we noticed that GREEN BY SOFTWARE is widely understood and easy to identify or create since it enables the positive impacts of Information Communication and Technology (ICT) area in sustainability, however, Green IN Software is still obscure and overlooked. So I emphasize on green in software by implementing SCRUM on green software engineering. It is possible to observe that studies related to this topic have a lack of application in the real-world scenario of software development. It is possible to find Sustainable Software Engineering Practices that contribute to software development in the academy and in the industry. The effectiveness of these practices needs to be applied during software development. The use of the theoretical framework for which the purpose is to serve as a guide during the application of these practices in real-world scenarios.

Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for Agile Software Development and the 12 Principles behind it. When you approach software development in a particular manner, it's generally good to live by these values and principles and use them to help figure out the right things to do given your particular context. One thing that separates Agile from other approaches to software development is the focus on the people doing the work and how they work together. Solutions evolve through collaboration between self-organizing cross-functional teams utilizing the appropriate practices for their context. scrum is founded on empiricism and lean thinking. Empiricism asserts that knowledge comes from experience and making decisions based on what is observed. Lean thinking reduces waste and focuses on the essentials.

Scrum employs an iterative, incremental approach to optimize predictability and to control risk. Scrum engages groups of people who collectively have all the skills and expertise to do the work and share or acquire such skills as needed. Scrum combines four formal events for inspection and adaptation within a containing event, the Sprint. These events work because they implement the empirical Scrum pillars of transparency, inspection, and adaptation[15].

The intended research is to address the issues of sustainability in development life cycle models by answering the following questions:

1. What are the sustainable measures we should take when applying SCRUM to software right from requirement phase to maintenance phase?

2. Can the existing models of agile methodology addresses the sustainable development of software in terms of five dimensions? If not, how to address them by new proposed model using Scrum?

The proposed research helps to develop model by using SCRUM, an agile methodology with the practice of the values contained in the Agile Manifesto and the 12 Principles behind the Agile Manifesto [8] and limitations in greensoft model[4].

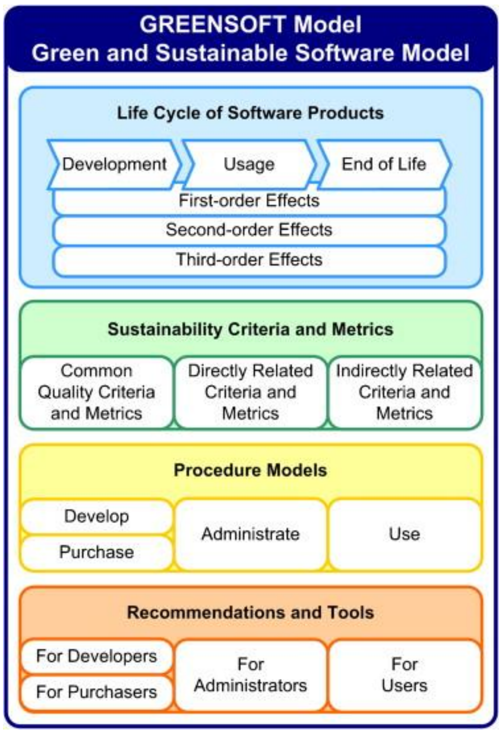

GREENSOFT Model
Green and Sustainable Software Model
Life Cycle of Software Products
Development
Usage
End of Life
First-order Effects
Second-order Effects
Third-order Effects
Sustainability Criteria and Metrics
Common Quality Criteria and Metrics
Directly Related Criteria and Metrics
Indirectly Related Criteria and Metrics
Procedure Models
Develop
Purchase
Administrate
Use
Recommendations and Tools
For Developers
For Purchasers
For Administrators
For Users

Fig -1: GREENSOFT MODEL

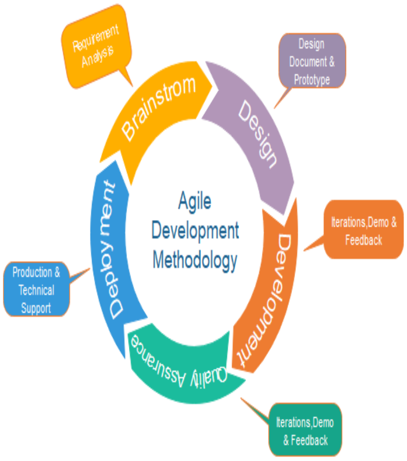

Requirement Analysis
Design Document & Prototype
Brainstrom
Design
Agile Development Methodology
Iterations,Demo & Feedback
Production & Technical Support
Deployment
Development
Quality Assurance
Iterations,Demo & Feedback

Fig. Agile Model

3. CONCLUSIONS

The main objective of our proposed GREEN SCRUM Model is to structure concepts, strategies, activities, and processes of Scrum model and especially of green and sustainable software and its engineering. This model acts as a reference model, which helps to organize and classify research results, actions, frameworks, process models, etc. using SCRUM. The model also suggests how SCRUM model of Software Development life Cycle (SDLC) can be enriched regarding sustainability.

REFERENCES

1. Harbla, A., P. Dimri., D. Negi & Y. S. Chauhan. Green computing research challenges: a review. International Journal of Advanced Research in Computer Science and Software Engineering 3(10): 1-13 (2013)

2. Sierszecki, S., K. Sierszecki., T. Mikkonen., M. Steffens., T. Fogdal., & J. Savolainen. Green software: greening what and how much?. Software, IEEE 31: 64-68 (2014).

3. Mahmoud, S.S. & I. Ahmad. A green model for sustainable software engineering. International Journal of Software Engineering and Its Applications 7(4): 55-74 (2013).

4. Naumann, S., M. Dick., E. Kern & T. Johann. The GREENSOFT Model: a reference model for green and sustainable software and its engineering. Sustainable Computing: Informatics and Systems 1(4): 294-304 (2011)

5. Shenoy, S. S., & R. Eeratta. Green software development model: an approach towards sustainable software development. India Conference (INDICON), India Annual IEEE, p. 1- 6 (2011).

6. Mahaux, M. & C. Canon. Integrating the complexity of sustainability in requirements engineering. In: First International Workshop on Requirements for Sustainable Systems. Belgium, p. 31-33 (2012).

7. Bozzelli, P., Q. Gu & P. Lago. A Systematic Literature Review on Green Software Metrics. VU University, Amsterdam (2013).

8. Dick, M., J. Drangmeister, E. Kern & S. Naumann. Green software engineering with agile methods. In: Green and Sustainable Software (GREENS), 2nd International Workshop, Netherlands, p. 78-85 (2013).

9. Sabharwal, M.R. Software power optimization analysis and optimization for energy-efficient software. In: Proceedings of the 17th IEEE/ACM international symposium on Low-power electronics and design, Fukuoka, p. 63-64 (2011).

10. Lago, P., Kazman, R., Meyer, N., Morisio, M., Muller, H.A., Paulisch, F., Scanniello, G., Penzenstadler, B. and Zimmermann, O. Exploring initial challenges for green software engineering: summary of the first GREENS workshop. at ICSE 2012. SIGSOFT Softw. Eng. Notes, 38(1), Jan. 2013, pp. 31-33.

11. Mohhankumar, M. and Anand Kumar, M. Green based Software Development Life Cycle Model for Software Engineering. Indian Journal of Science and Technology, vol. 9(32), 2016, pp. 1-8

12. Nasir Rashid, Siffat Ullah Khan, Agile practices for global software development vendors in the development of green and sustainable software, Journal of Software: Evolution and Process, 10.1002/smr.1964, **30**, 10, (2018).

13. Rashid, N, Khan, SU. Using agile methods for the development of green and sustainable software: Success factors for GSD vendors. *J Softw Evol Proc*. 2018; 30:e1927. https://doi.org/10.1002/smr.1927

14. Penzenstadler, Birgit. "Infusing Green: Requirements Engineering for Green In and Through Software Systems." *RE4SuSy@RE* (2014)

15. Schwaber, Ken; Sutherland, Jeff (November 2017), The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game, retrieved May 13, 2020