APPLYING DESIGN THINKING AND SUSTAINABILITY ASSESSMENT FRAMEWORK FOR REQUIREMENTS ENGINEERING OF RESPONSIBLE SOFTWARE

Dr. Rasika S Mallya, Associate Professor, MCA Department, DES's NMITD, Mumbai-28.

Abstract
Effective requirements engineering is one of the crucial way to build sustainable software. Design thinking provides some of the strategies and toolbox for effective requirements engineering. This paper studies design thinking strategies applicable for various requirements engineering activities. This paper also identifies some of the techniques useful for sustainable software development. To this aim, sustainability assessment framework toolkit is used to identify quality attributes of software. Thus, this study explores design thinking strategies and Sustainability assessment framework for Requirements Engineering of sustainable software.
Keywords: Requirements Engineering, Design Thinking, Sustainable Software, Sustainability Assessment Framework.

Introduction:
Responsible software is a part of sustainable ICT used for responsible enterprises which integrate social, environmental, ethical, human rights and consumer concerns into their business operations. Responsible software can play vital role in enhancing the social and environmental responsibility of enterprises through efficient resource management, transparency and accountability, sustainable product design and renewable energy management. [1]
Design thinking is human centred approach to innovation with the toolkits or methods. Design Thinking is interpreted in three ways: 1. A process with a sequence of steps according to process framework 2. A collection of toolkits or methods to design this framework 3. A mindset with human-centered principals to implement process framework with the help of toolkit. Design thinking deals with both problem space and solution space with phases as
Problem Space: 1. Problem Definition, 2. Need finding 3. Synthesis
Solution Space: 1. Ideation 2. Prototyping 3. Testing.
Design thinking is used as a collaborative approach to develop innovative digital solutions. [2]
Design thinking can enable sustainable software development by minimizing the environmental effects of software development at all its lifecycle stages. Sustainable software practices are opting for energy-efficient programming languages, developing a user-friendly digital experience with efficient accessibility and performance optimization. These practices lead to green software engineering. This emerging discipline leads to development of environmentally-friendly software through which green software engineers can reduce the carbon footprint of their applications. Software engineering for sustainability supports all dimensions of sustainability – human, social, economic, environmental and technical. [3] The social sustainability assures that the software intensive systems shall have support of social communities and the processes of software intensive systems shall benefit the social communities. The environmental sustainability assures that the software intensive systems shall have energy efficient resources and they shall create ecological awareness to be environmental sustainable. The technically sustainable software systems will have a long term use adapting the dynamic execution environment. The economically sustainable software systems will have preserved capital value and financial value. [4]

This paper focuses on requirements engineering of sustainable software with some design thinking strategies and applying Sustainability Assessment Framework to software to identify related requirements.

The structure of this paper is as follows: Section II gives brief about artefact base requirements engineering, Sustainability Assessment Framework, Design Thinking and Strategies and Sustainable Wearable Healthcare Application. Section III applies Sustainability Assessment Framework for Sustainable Wearable Healthcare Application. Section IV concludes the research work.

Literature Review:

Artefact-based Requirements Engineering (ABRE): AMDiRE:

An Artefact-based Requirements Engineering (ABRE) Reference Model is a structured approach to requirements engineering that focuses on the creation, management, and utilization of artefacts. Artefacts are deliverables produced during the software development process, such as documents, models, diagrams, or any other deliverable that captures requirements.

Implementation Steps for an ABRE Reference Model:

1. **Define Artefacts**: Identify the types of artefacts needed for the project and define their structure and content.
2. **Establish Processes**: Set up the processes for eliciting, analyzing, specifying, validating, and managing requirements.
3. **Assign Roles**: Define the roles and responsibilities of stakeholders, requirements engineers, and project managers.
4. **Select Tools**: Choose appropriate tools for modeling, collaboration, documentation, and traceability.
5. **Develop Templates**: Create templates for different types of artefacts to ensure consistency and completeness.
6. **Train Team**: Provide training to the project team on the ABRE model, tools, and processes.
7. **Execute and Monitor**: Implement the requirements engineering processes, produce the artefacts, and continuously monitor and improve the process.

AMDiRE model consists of around 70 artefacts which represent functional as well as non-functional requirements. The artefacts are classified as follows:

1. **Context Specification:** This class represents project scope, stakeholder model, goals and domain models.
2. **Requirements Specification:** This class represents system vision – covers basic idea of application, usage model to represent functional requirements and service model to represent services offered by application.
3. **System Specification:** This class represents components to describe internal behaviour of components and data model to represent entities. [5]

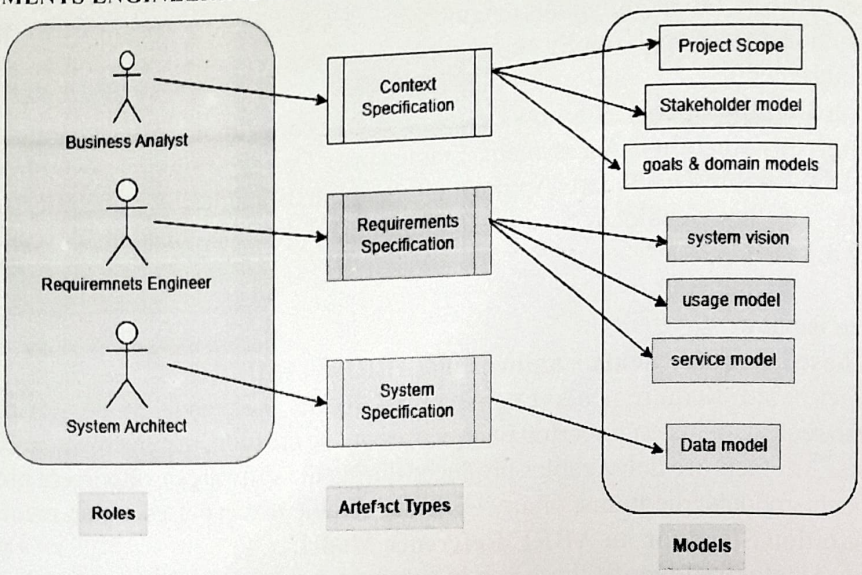

Figure 1: Specification wise artefacts for requirements engineering

Design Thinking for Requirements Engineering:

Many projects suffer due to proper requirements engineering or dynamic requirements. Design thinking can be one of the tool to make requirements engineering effective with interdisciplinary teamwork and fast and non-technical prototyping cycles. Design thinking process consists of following phases:

1. **Redefine:** The initial point of design thinking where stakeholder's engagement and their impact and challenges will be assessed.
2. **Needfinding and Synthesis:** Design thinking team implements research techniques to identify hidden and unexplored needs. Problem framing and reframing helps to identify the exact requirements.
3. **Ideation:** This activity focuses on creating ideas to simulate thinking and identify more appropriate requirements.
4. **Prototyping:** This is the most effective way of representing solution needs and to facilitate communication with end users via feedback.
5. **Testing:** Based on User's feedback, improvement opportunities are explored.

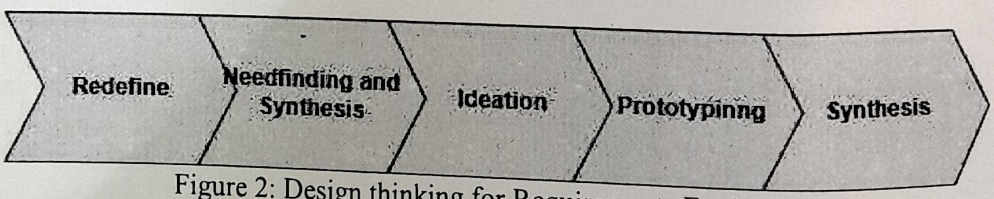

Figure 2: Design thinking for Requirements Engineering

Following are the strategies of design thinking which can be integrated with requirements engineering [7]:

	Upfront Design Thinking	Infused Design Thinking	Continuous Design Thinking
Features	Used during the initial phase of a project to clarify	Leveraged within an existing Requirements	Continuous implementation of design thinking and

	ambiguous requirements and establish a clear vision for the solution.	Engineering (RE) process to generate new insights	RE techniques to identify user needs from vision till functional requirements
Result	Clear Solution Vision in form of mock up	New unexplored requirements identification with specific artefacts	Software requirements specifications tracing to customer needs
Benefits	Deep context understanding can be achieved	Few changes are required in existing RE process, easy adaptation with Design thinking	Seamless integration, Human-focused perspective throughout project

Design thinking is the collection of design methods which can be implemented with Design thinking strategies. Following are some of the design methods which can be used in each phase of design thinking along with specific strategy. The design thinking toolbox offers flexible way of implementing design thinking with requirements engineering. [6][8]

Sr. No	Phase Name	Goal of the Phase	Tools/ Methods	Design thinking strategy
1	Empathize	Research User needs	Interviews, observations and active listening	**Upfront Design Thinking**
2	Define	State user needs and problems	Desk Research, framing and reframing, stakeholder mapping	**Infused Design Thinking**
3	Ideate	Challenge assumptions and create ideas	Questions, brainstorming, brain writing	**Infused Design Thinking**
4	Prototype	Start to create best possible solutions	Paper prototype, role playing, sketches	**Continuous Design Thinking**
5	Test	Testing of prototype	Feedback capture grid	**Continuous Design Thinking**

The above strategies along with design thinking toolbox can be used to identify requirements of sustainable software.

Sustainable Software Development:

The Sustainable Software Development minimizes environmental effects of software development at all life cycle stages such as optimizing code to reduce energy consumption, using more efficient hardware, and opting for energy-efficient data centers, less downtime and quicker updates. [9] Green coding and green programing are the techniques used for Sustainable Software Development. Green coding emphasizes minimizing the use of resources

and reducing code complexity, Sustainable Software Development ensures practices like green architecture, user friendly digital experience, AIML models giving target accuracy, maximum usage of cloud, IT infrastructure with minimal emissions and electronic waste. Following principles of design thinking can be used to software sustainable:

I. **Reduce complexity** – code complexity, data migration complexity,

II. **Identify unintended negative consequences:** Social negative impacts can be identified and mitigated inn advance.

III. **Be inclusive and clear:** Software usability shall be for all age group users with simple, clear and unambigous language.

IV. **Reduce waste:** Reusing existing design patterns, [10]

The sustainable software or green software has overall impact on business as reduction in energy cost, simplification and streamlining complex infrastructure. Thus, finally these practices contribute for financial success.

Sustainable Wearable Healthcare Application:

This research paper considers sustainable ubiquitous software suitable for wearables to monitor body parameters for well being of human. The research in wearable domain has attempted to improve quality of life from sustainability viewpoints. Along with monitoring and measuring physical activities, sustainable wearables can deal with domains as human data tracking, human data analysing, ambient intelligence with supporting middleware. [12] The wearable technology with sustainable values can be viewed as shown in figure:

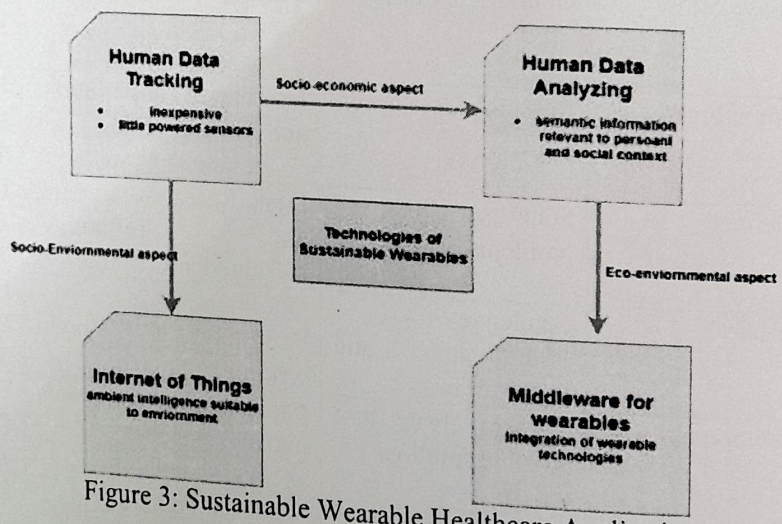

Figure 3: Sustainable Wearable Healthcare Application

Applying SAF for Sustainable Wearable Healthcare Application:

This section applies Sustainability Assessment Framework toolkit to wearable healthcare application. This framework provides set of instruments to support software architect and design to make the software sustainable. Sustainability Assessment Framework toolkit mainly consists of two main instruments – Sustainability-Quality model (SQ model) and Decision Map (DM). Both the instruments adopt sustainability dimension.

To frame and design sustainability relevant designs and quality requirements, the Decision Maps are used visually. Decision Maps form a framework to capture, reason and uncover dependencies among sustainability related quality requirements. This framework can be represented in visual diagram. [11] The notations for Decision Map are shown in below figure:

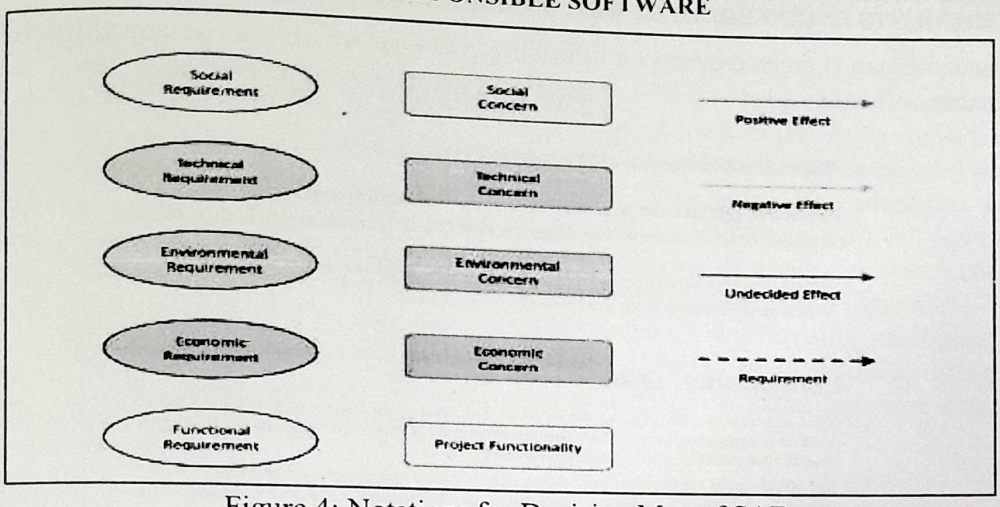

Figure 4: Notations for Decision Map of SAF

In above notations, oval shape represents requirements and rectangle shape represents sustainability concern related to requirements.

Sustainability Assessment Framework toolkit helps to create Decision Maps for specific software intensive system. The instruments such as checklist and decision graph create Decision Maps. The checklist helps to create Decision Maps in systematic way with identification of Quality Attributes which are relevant and important for design of a software.

Sustainability-Quality model (SQ model) represents collection of quality attributes based on four sustainability dimensions – social, environmental, technical and economic sustainability. While all four dimensions are to be considered during design phase, it is not always necessary for every dimension to be relevant for specific software. [13]

Based on notations of Sustainability Assessment Framework, following Decision Map is created with design concerns of sustainable wearable application for health monitoring. As shown in below decision map, green rectangle shows environmental concern, blue rectangle shows technical concern, white rectangle shows main functionality of the project, yellow rectangle shows societal concern. etc.

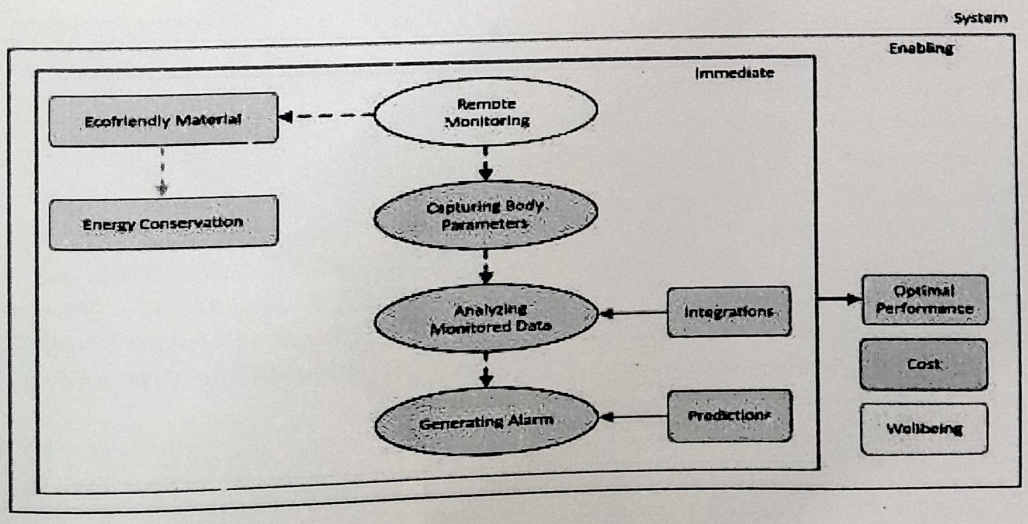

Figure 5: Decision map of Sustainable Wearable Healthcare Application

The decision map is created based on following checklist which is a constant checklist for any sustainable software.

Sustainability-Quality concerns

Identification of sustainability-quality attributes(QAs).
Check the box once the question has been answered.

- ☐ What are the most relevant sustainability dimensions?
- ☐ What are the key QAs you need to address?

Identification of expected effects

- ☐ (for each identified QA) what are the possible inter-dependent QAs?. Use the suggestions in the Dependency Matrix. You can start with the most relevant sustainability dimension.
- ☐ for each pair of interdependent QAs, draw an arrow and assign the expected effect (positive, negative, undecided). Use the suggestions in the Dependency Matrix, if applicable.

Update the list of identified QAs

- ☐ (for each interdependent QA) are then additional QAs that we should consider?

Figure 6: Checklist for SAF

The implications of above study is shown in below figure. Due to application of design thinking strategies make the requirements "desirable" which focuses on human and social concern of sustainability, the solution becomes "feasible" which focuses on technical and environmental concern of sustainability and design thinking strategies also make the solution "viable" which identifies economical concern of sustainability. The sustainability concerns are color coded as per sustainability assessment framework.

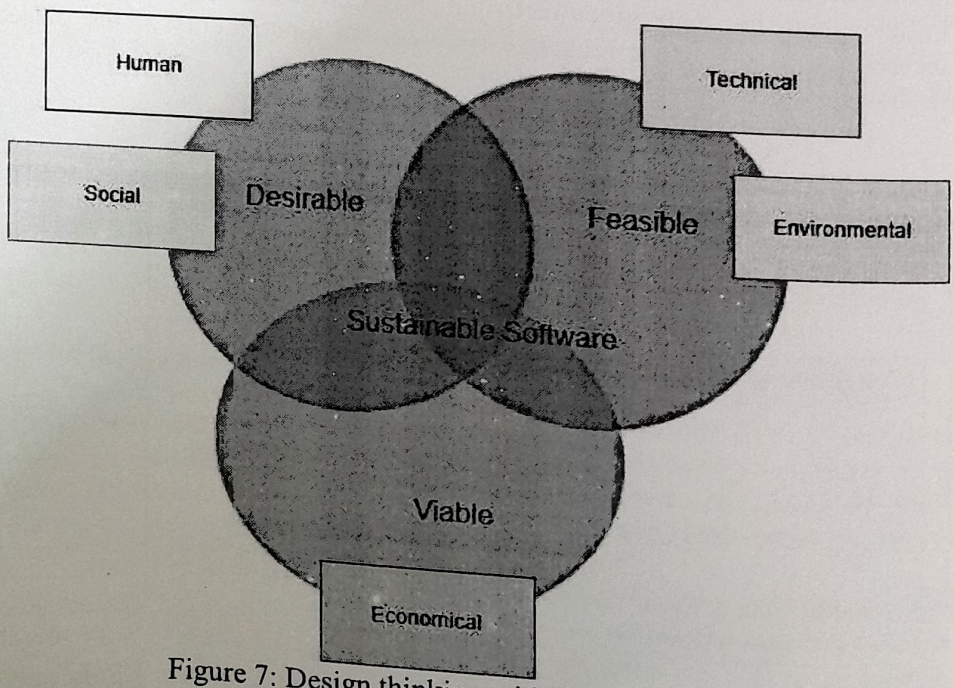

Figure 7: Design thinking with sustainable values

Conclusion:

In this paper, we present design thinking strategies useful for requirements engineering. Also, design thinking tools are explored for various requirements engineering activities. Design thinking strategies and sustainability assessment framework toolkit is integrated and applied for requirements engineering of wearable software. This paper also identifies features of sustainable wearable software. The sustainability assessment framework along with design thinking strategies represent the quality attributes in terms of quality requirements to make software sustainable. With the help of sustainability assessment framework, this study identified Decision Map of sustainable wearable software with quality attributes as energy conservation, ecofriendly material, well-being etc. In similar way, sustainability assessment framework and design thinking strategies can be applied to make software sustainable or responsible.

References:

1. Sergio España, Sjaak Brinkkemper, Responsible software, 4th International Conference on ICT for Sustainability (ICT4S 2016), © 2016. The authors - Published by Atlantis Press
2. Jennifer Hehn, The Use of Design Thinking for a Human-centered Requirements Engineering Approach.
3. https://beetroot.co/greentech/best-practices-of-sustainable-software-development/
4. PATRICIA LAGO, SEDEF AKINLI KOÇAK, IVICA CRNKOVIC, AND BIRGIT PENZENSTADLER, Framing Sustainability as a Property of Software Quality, COMMUNICATIONS OF THE ACM, OCTOBER 2015 | VOL. 58 | NO. 10
5. D. M´endez Fern´andez, B. Penzenstadler, Artefact-based Requirements Engineering: The AMDiRE Approach, SPRINGER. PREPRINT
6. Jennifer Hehn, Daniel Mendez, Combining Design Thinking and Software Requirements Engineering to create Human-centered Software-intensive Systems,
7. Jennifer Hehn, Daniel Mendez, Falk Uebernickel, Walter Brenner, Manfred Broy, "On Integrating Design Thinking for a Human-centered Requirements Engineering", Published by the IEEE Computer Society
8. https://www.interaction-design.org/literature/topics/design-thinking#what_is_design_thinking_(dt)?-0
9. https://medium.com/kainosxd/green-software-how-design-thinking-can-help-develop-sustainable-software-87ad54190251
10. https://beetroot.co/greentech/best-practices-of-sustainable-software-development/
11. Patricia Lago, Nelly Condori Fernandez, Iffat Fatima1,Markus Funke1, Ivano Malavolta, The Sustainability Assessment Framework Toolkit: A Decade of Modeling Experience
12. Gurova Olga, Timothy Robert Merritt, Eleftherios Papachristos and Jenna Vaajakari, "Sustainable solutions for wearable technologies: Mapping the product development lifecycle", Sustainability 2020, 12, x; doi: FOR PEER REVIEW www.mdpi.com/journal/sustainability
13. Jaewoon Lee, Dongho Kim, Han-Young Ryoo and Byeong-Seok Shin, "Sustainable Wearables: Wearable Technology for Enhancing the Quality of Human Life", Sustainability 2016, 8, 466; doi:10.3390/su8050466 www.mdpi.com/journal/sustainability