Karlskrona Manifesto: Software Requirement Engineering Good Practices

Shola Oyedeji and Birgit Penzenstadler

Introduction

- Sustainability has become one of the major issues of society today because of the impact of human activity on our planet.
- This includes interactions between individuals, within communities, and between companies and users in which software plays a key role in connecting everyone.
- Therefore its important to design, develop and maintain a sustainable software.
- The Karlskrona Manifesto for Sustainability Design serves as a guiding principles upon which processes, methods and tools can be developed for software systems.
- This will support the design and development of sustainable software just like other manifestos in the history of computer science and software engineering.

Goal

- Explores the transition of Karlskrona Manifesto Principles for Sustainability Design into processes, tools and methods that can educate and encourage software system requirement engineers, designers and developers in eliciting software sustainability requirements.

- Manifestos like the Agile manifesto are one example that has transitioned into processes, methods and tools to help practitioners using Agile in software development

Processes and Guiding Reasons behind Karlskrona Manifesto principles

- **Principles** not techniques as a guide for building, developing and improving new/old techniques and tools to support sustainability design.
- Provide a broader **scope** to be all-inclusive and encompassing all aspects of sustainability.
- Bottom up approach to cover all **emerging structure** from contributions of all participants involved in the initiation of the manifesto.
- Discussion through **participation and transparency** to encourage broader engagement of different experts of sustainability and interested participants.
- **Conversation over consensus** to enable dialogue among the community of stakeholders and all interested participants.
- **Minimal and adaptive process** focussed on emergent content and structure.
- **Synchronous collaboration**. Contents of the manifesto were written through synchronous collaboration.
- **Iterative evolution**. A common vision was formulated to guide the incremental evolution of the manifesto.

Karlskrona Manifesto Principles

Principles	Description
P1. Sustainability is systemic	Sustainability is never an isolated property
P2 Sustainability has multiple dimensions	Inclusiveness of all the different dimensions of sustainability
P3. Sustainability transcends multiple disciplines	Addressing the challenges from multiple perspectives
P4. Sustainability is a concern independent of the purpose of the system	Sustainability has to be considered even when system primary goal is not sustainability
P5. Sustainability applies to both a system and its wider contexts	Consider sustainability of the system itself and how it affects the sustainability of the wider system environment in which it operates
P6. System visibility is a necessary precondition and enabler for sustainability design	Visible system status and enable participation and informed responsible choice
P7. Sustainability requires action on multiple levels	Seek interventions that have the most leverage on a system and consider the opportunity costs
P8. Sustainability requires to meet the needs of future generations without compromising the prosperity of the current generation	Innovation in sustainability can play out as decoupling present and future needs
P9. Sustainability requires long-term thinking	Multiple timescales, including longer-term indicators in assessment and decisions should be considered

Karlskrona Manifesto Principles & SDLC Phases

SDLC Phases	Karlskrona Manifesto Principles
Phase 1. Project Definition	P1- This ensures that the project initiation considers sustainability in the overall project definition from the beginning. P2- Software sustainability has different dimensions that have to be taken into account from the beginning for better project management with different stakeholders. P3- Software projects usually involve stakeholders from different domains, incorporating their sustainability concerns provides better management of those concerns from multiple perspectives which can help the incorporation of sustainability for the software.
Phase 2. User Requirements Definition	P2- Recording and documenting user feedback on their perception of sustainability during requirements elicitation will foster better sustainability analysis during the system analysis and design phase.
Phase 3. System Requirements Definition	P4- During elicitation of system requirements to consider sustainability concerns for the system during the requirements definition even when it is not a core part of the user requirements. P5- Cross evaluate the consequential impacts of the system sustainability requirements and the environment in which the system will function.
Phase 4. Analysis and Design	P2- Applying this principle provides a blueprint for system evaluation from all sustainability dimensions (Economic, environment, social, individual and technical). P4- This principle provides a rethink of how to conduct analysis of system design with consideration of sustainability in order to facilitate development of sustainable system. P6- Application of this principle enables better visual and visible overview of the system from different levels of abstraction. P8- This will provide better understanding during analysis to make better choices that will help the potential users of the system in present and in future when the system evolves.
Phase 5. Development	P2- This will encourage developers during this phase to consider different sustainability dimensions especially technical, social and individual dimensions P4- Encourages the search for better avenues to make the system sustainable from the development perspective (developers) and also the functions of the system to aid longevity.
Phase 6. Integration and Testing	P2- Provides integration and test team to have a sustainability template that can be used to test the system for all sustainability dimensions based on the sustainability requirement output from phase 2, 3, and 4. P4- Application of this principle will aid consideration of sustainability in this phase even if the primary focus of system is not about sustainability.
Phase 7. Implementation	P5- Provides a beforehand reasoning for the development team to consider sustainability of the system, its production environment and when push live for use. P7- The use of this principle will aid consideration of seeking the involvement of different stakeholders to make the actualization of the system sustainability possible in the production environment and when pushed live.
Phase 8. Sustainment / Maintenance	P9- At this stage, this principle helps to create the conscious awareness so that when the system is in live environment, there will be continuous evaluation to assess the system sustainability and think of ways for optimizing and improving sustainability of the system from the different dimensions.

Requirement Engineering and Karlskrona Manifesto Principles

- Requirements engineering as a starting point in any software development has a crucial role to play in exemplifying the use of the manifesto principles in software systems requirements elicitation, engineering and development.

- However, the core challenge is how to exemplify these principles through practical application in software development and document it.

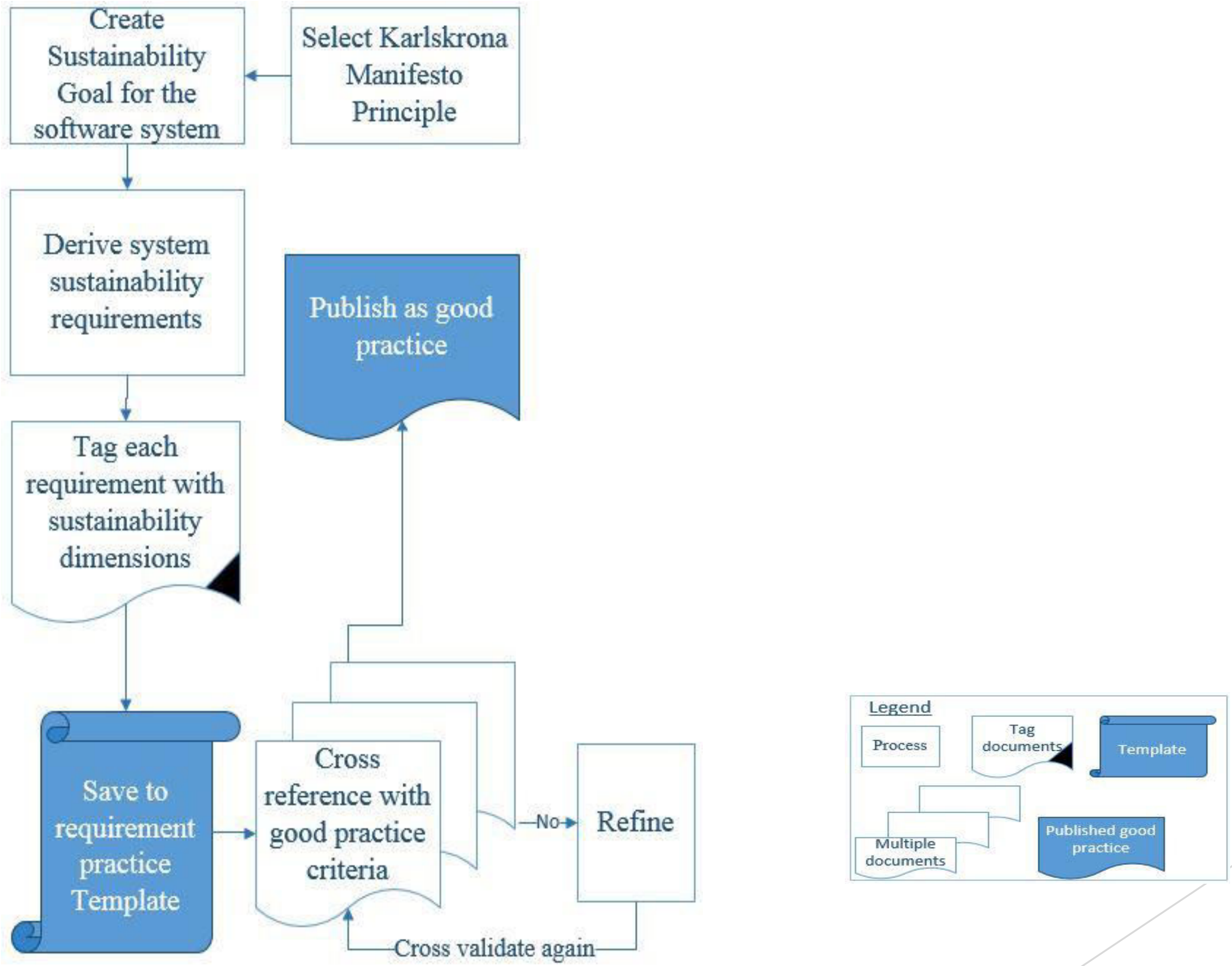
- This will:
  - Increase sustainability awareness amongst those interested in software systems design and development
  - Provide more information on how these principles have been applied in different systems
  - Offer feedback on how to better translate these principles into tools, methods and processes.
  - Provide novice requirement engineer and software developer a starting point for engineering sustainability into software requirement and development.

- Therefore a best practice documentation for using the Karlskrona Manifesto principles in requirement engineering will serve as a pivotal template for all stakeholders in software system development.

Best Practice Documentation and Templates

- A “best practice” (BP) is a practice that is not only good but has proven to work well and produce good results and therefore is recommended as a model.
- According to Schatten, a BP is the transfer of knowledge based on years of success, mistakes and failures from experienced developers to novice developers.
- These BP can be some good (patterns) and bad decisions (anti-patterns) from concrete projects that are presented as abstracted scenarios.
- Designing and developing well-structured sustainable software is a challenge especially for young and novice developers.
- Such challenges can be eased for them with knowledge of how best to develop a well-designed software systems from proven procedures based on BP.

METHOD FOR DOCUMENTING SOFTWARE SUSTAINABILITY REQUIREMENTS BEST PRACTICE



Select Karlskrona Manifesto Principle

Create Sustainability Goal for the software system

Derive system sustainability requirements

Tag each requirement with sustainability dimensions

Save to requirement practice Template

Cross reference with good practice criteria

No

Refine

Cross validate again

Publish as good practice

Legend

Process

Tag documents

Template

Multiple documents

Published good practice

Validation of Best Practice Requirement

- **Effective and successful:** A “good practice” has proven its strategic relevance as the most effective way in achieving a specific objective; it has been successfully adopted and has had a positive impact on individuals and/or communities.

- **Environmentally, economically and socially sustainable**: A “good practice” meets current needs, in particular the essential needs of the world’s poorest, without compromising the ability to address future needs.

- **Technically feasible:** Technical feasibility is the basis of a “good practice”. It is easy to learn and to implement.

- **Inherently participatory**: Participatory approaches are essential as they support a joint sense of ownership of decisions and actions.

- **Replicable and adaptable**: A “good practice” should have the potential for replication and should therefore be adaptable to similar objectives in varying situations.

- **Reducing disaster/crisis risks, if applicable**: A “good practice” contributes to disaster/crisis risks reduction for resilience.

DESCRIPTION OF TEMPLATE FOR SOFTWARE SUSTAINABILITY REQUIREMENT ELICITATION BEST PRACTICE

Element	Description
Title	Which title best describes the best practice?
Date	What month and year is the “good practice” published or documented?
Authors	Who wrote the good practice document?
Target Audience	Who is the target group? To whom is this document useful?
Objective	What is the goal or aim of the best practice?
Location	What is the geographic location in which this practice can be applied for software system (country, region, town or village)? Examples: system for a country’s, state, province health care system or banking system or a commercial software application
Stakeholders	Beneficiaries of this best practice? Who are the users, institutions and implementing agencies of the best practice?
Methodology	What methodology was used in documenting the best practice? What were the process steps involved?
Selected Karlskrona manifesto principles	What are the principles that served as guide for creating the best practice for requirement elicitation?
Requirements	What were the requirements used in the best practice? How was sustainability considered in the requirement?
Validation	How was the best practice validated? Did the best practice fulfil the best practice criteria?
Impact	What there an impact in the application of the best practice?
Lessons Learnt	What are the key take away from the application the best practice?
Sustainability	What are the dimensions of sustainability covered in the best practice application?
Contact Details	What is contact details of those responsible for the best practice?

TEMPLATE OF SOFTWARE SUSTAINABILITY REQUIREMENT ELICITATION BEST PRACTICE

Element	Description
Title	Sustainability user awareness best practice of online hospitality service for short term house renting and sharing
Date	11-06-2018
Authors	Shola Oyedeji, Birgit Penzenstadler
Target Audience	Requirement engineers, Web developers, Business analyst
Objective	Document best practice in requirement elicitation for a web system in order to: • Create awareness among web application developers on how to elicit sustainability requirements • Encourage development of web systems with consciousness of sustainability for end users while using the web application
Location	Applicable worldwide for any web system
Stakeholders	Software requirement engineer, Programmers and Business analyst
Methodology	• Discussion among software development team on what sustainability means to them by going through the Karlskrona manifesto principles • Use the Karlskrona manifesto principles as guide during requirement elicitation during discussion with the end user with aid of the sustainability analysis chat • Record all the requirements in the user requirement specification (URS) and software requirements specification (SRS) • Dialogue about which requirements can better influence end user awareness about sustainability in the user and software requirements specification (URS and SRS) document. • Selected identified requirements • Discussion between with the requirement engineer, end user and programmers about these sustainability requirements to see if implementation is possible or if there is need for modification • Modify requirement in URS and SRS with a set of new requirements targeted towards sustainability based on discussion between the requirement engineer, end user and programmer
Selected Karlskrona manifesto principles	Principle 2: Sustainability has multiple dimensions Principle 6: System visibility is a necessary precondition and enabler for sustainability design Principle 7: Sustainability requires action on multiple levels
Requirements	Functional Requirement REQ 1 –Registration (user must be able to register using web form and receive a notification via email) • Sustainability requirement added to this general registration requirement is to include short sustainability tips/links in the registration notification email such as how to recycle common grocery items, use home energy, water, heater and nearest cycling station for getting bicycle commuting REQ 2- UI Search Results (Display search results for all homes with prices and availability to users) • The requirement for sustainability added to this search requirement is to include the CO2 emission for all homes based on the user (searcher location) to the search home (destination) and also add green level label for all homes based on user feedback on how easy to recycle, access to path way for walking or bicycle or public transportation and energy usage during their stay in a home Non-Functional Requirements REQ 3 – Performance (ensure good response time) • The sustainability consideration for this requirement is write good compact design codes during development that can determine the exact CPU usage for specific components of the web application and optimize them for less CPU usage • Create effective and efficient algorithm for data structures to help use minimum system resource which can in turn improve respond time and reduce application energy usage
Validation	Programmer, Business analyst and requirement engineer cross validate those requirements with the best practice criteria
Impact	Promote sustainability awareness among software developers and end users Provide opportunity to rethink how software requirement are elicited with consideration of sustainability
Lessons Learnt	1. Software developers don’t like too much documentation, so this template has been simplified 2. Requirement engineers appreciated the mapping of Karlskrona manifesto with software development phases 3. Software developers said they would appreciate more documentation on software sustainability for agile development process though they find the mapping in Table 3 useful for them to understand how each of the Karlskrona manifesto relates to each of the software development phases 4. Developers started discussing about coming to office by bicycle or public bus transport instead of their car to reduce CO2 emission
Sustainability	The requirements in this template covers: Social Sustainability Environment Sustainability Individual Sustainability
Contact Details	shola.oyedeji@lut.fi , birgit.penzenstadler@csulb.edu

Background of Expert Evaluators for Best Practice Template

Expert	Background	Company Type	Years of Experience
1	Software Tester	Software Development	5
2	Requirement Engineer	Software Development	3
3	Programmer	Software Development	4
4	UI Designer	Software Development	3
5	Business Analyst	Software Development	3
6	Software Developer	Software Development	4
7	Programmer	Software Development	3
8	IT Manager	Software Development	4
9	CEO / Software Developer	Startup Software Development	3
10	ICT Engineer	Telecom	4
11	Programmer	Finance	3
13	Product Tester / Integration Engineer	HR	3
14	Project Manager / UX expert	Software Development	4
15	Not Provided	Not Provided	Not Provided

Discussion

- The systematic mapping of the Karlskrona Manifesto to SDLC phases aids requirements engineers and software developers in understanding how the Karlskrona Manifesto for software sustainability design relates to the software development life cycle

- There is a lack of methodological support for sustainability design in requirement engineering because it is not part of most companies practice.

- The method presented in this paper serves as support for helping requirements engineers, software developers and all stakeholders in documenting best practices from sustainability design in requirements engineering using a structured method.

- Documenting best practices using the proposed template presented here educates and promotes awareness among those involved in the requirements engineering process of software development and all stakholders.

- This can be one way of persuading them to see benefits of eliciting software requirements and developing software system in a new way with support for sustainability design.

Conclusion and Future Work

- Exemplifying the use of Karlskrona Manifesto Principles for Sustainability Design through best practice can create more awareness and encourage creation, design and development of new tools, methods and processes for software sustainability requirement elicitation and design.

- The best practice Template can serve as a practical way to educate both experience and novice requirement engineers and developers on how to gather requirements for sustainable software design and development.

- Future work includes the application of the proposed methodology in industrial case studies and using the template to document best practices from those case studies.

- Specifically, during the evaluation, the expert group requested a mapping of the Karlskrona Manifesto to agile software development method, especially to Scrum.

Thank You

Questions ?