Sustainability Profiling of Long-living Software Systems

Ahmed D. Alharthi, Maria Spichkova and Margaret Hamilton
RMIT University, Melbourne, Australia
Email: {ahmed.alharthi, maria.spichkova,margaret.hamilton}@rmit.edu.au

***Abstract*—This paper introduces a framework for software sustainability profiling. The goal of the framework is to analyse sustainability requirements for long-living software systems, focusing on usability and readability of the sustainability profiles. To achieve this goal, we apply a quantitate approach such as fuzzy rating scale-based questionnaires to rank the sustainability requirements, and the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) to analyse the results of questionnaires and to provide a basis for system profiling.**

The core profiling elements provided by our framework are (1) a sustainability five-star rating, (2) visualisation of the five sustainability dimensions as a pentagon graph detailing combination for individual, social, technical, economic and environmental dimensions, and (3) a bar graph of overall sustainability level for each requirement. To ensure sustainability, the proposed profiling framework covers the five dimensions of sustainability to quantify the sustainability of any software system not only during the requirement gathering phase but also during maintenance phase of software system lifecycle.

I. INTRODUCTION

Addressing the impacts of software systems on sustainability is a first-class quality concern beside usability, safety and security [1]. A number of studies showed that if a software system is developed without taking sustainability requirements into account, the system could have negative impacts on individual, social, technology, economic, and environment sustainability, cf. [2]–[5]. Environmental awareness is crucial for software engineering, especially in the case of large-scale systems having many thousands of users.

The analysis of system sustainability has to be initiated on the requirements engineering (RE) phase [6], [7]. Based on this idea, Becker et al. [8] emphasised that the importance of identifying stakeholders whose outside interests are affected, and the use of long-life scenarios techniques during requirements elicitation could forecast potential impacts. Duffy [9] highlighted that sustainability could be achieved especially in the social dimension through usability, which is a non-functional requirement, and its traditional methodologies.

This question is especially important for long-living systems, where the stakeholders requirements and preferences might change over the time the system is in use. For example, a system that can be seen as sustainable today, might be rated as environmentally unsustainable in few years, while new techniques to increase environmental sustainability are developed. To solve this problem, we require an easy-to-use profiling framework based on quantitate approaches that would allow to analyse the up-to-date system sustainability profiles, based on system characteristics and the up-to-date ratings (quotations) of the corresponding requirements. Usability and readability of the approach is crucial to make it applicable for real software development processes, as the quotation process and the generated profiles have to be easy-to-use by all stakeholders.

Contributions: To ensure the sustainability of long-living software systems over their eitire live-cycle, we propose a framework for sustainability profiling. The framework allows to analyse sustainability requirements for long-living software systems. The up-to-date profiles could be generated both during the RE and the maintenance phase of the software system lifecycle. The framework workflow is presented in Figure 1. First of all, stakeholders are assigned to a group to rate requirements from the different perspective of sustainability dimensions (individual, social, technical, economic and environmental). Then, a fuzzy rating scale is used to avoid imprecision for answering quantitative questionnaires [10]. As the next step, the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS, cf. [11]) is utilised to find alternatives that are the nearest distance from the positive ideal solution and farthest distance from the negative ideal solution. The software sustainability profiling includes an overall picture of how sustainable a software system really is. The profile is presented as three core elements: (1) a five-star rating, (2) five dimensions of sustainability in a pentagon graph, and (4) an overall measure of sustainability for each requirement in a bar graph.

Outline: The rest of the paper is organised as follows. In Section II we discuss the background and related work. Section III introduces our framework for software sustainability profiling. Section IV introduces an example scenario to show how the framework can be used to profile software systems. Section V summarises the core contributions of our work.

II. BACKGROUND AND RELATED WORK

In this section we discuss the research directions and approaches that provide a background for our framework: RE for sustainable systems, the idea of the sustainability profiling, quantitative approaches, approaches using the fuzzy rating scale, and the TOPSIS framework for requirements analysis. We selected TOPSIS for our sustainability profiling framework, as this technique has been successfully used for prioritising requirements and solving conflict among non-functional requirements, cf. [11]–[13]. Previously, TOPSIS was used

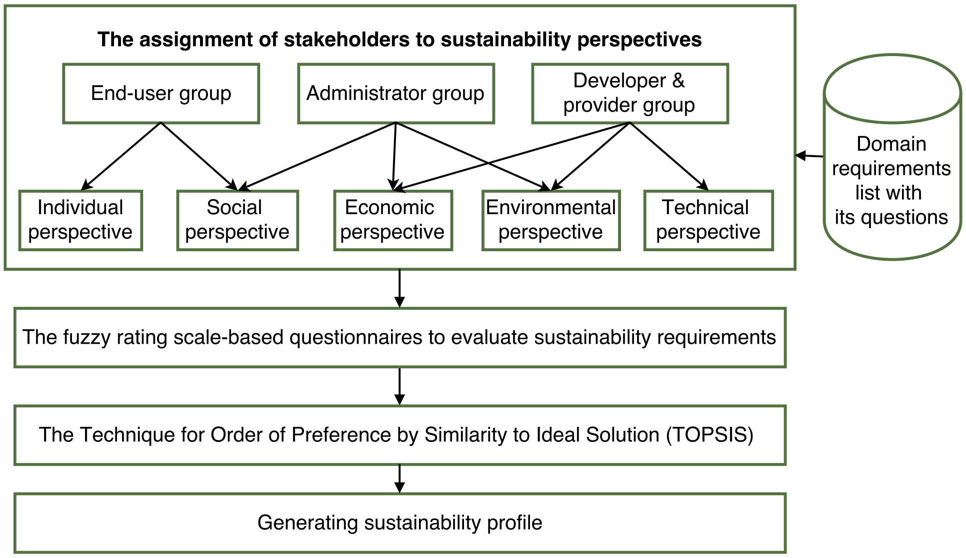

Fig. 1. Software Sustainability Profiling Framework

without taking into account the sustainability aspects, but the extension to evaluate sustainability requirements is possible and easy to implements. In the sustainability dimensions we have the same kind of relations among requirements: (1) each requirement has impacts on other requirements, and (2) each requirement has positive or negative impacts on sustainability dimensions that could be maximised or minimised during the TOPSIS procedure.

A. *Requirements engineering for sustainable systems*

The RE phase of software development focuses on discovering, developing, tracing, analysing, qualifying, communicating and managing system requirements, cf. e.g., [14]. Lami et al. [15] proposed to define a sustainable software process as one which meets realistic sustainability objectives, taking into account not only direct but also indirect impacts of the software on economy, society, human beings, and environment.

Penzenstadler [16] defined RE for sustainability as *"the concept of using requirements engineering and sustainable development techniques to improve the environmental, social and economic sustainability of software systems and their direct and indirect effects on the surrounding business and operational context"*.

Sustainability in software has various dimensions. Goodland [17] suggested to distinguish the following four dimensions: *human (individual), social, economic and environmental sustainability*. Penzenstadler and Femmer [5] as well as Razavian et al. [18] added to the new dimension of *technical sustainability*.

In our framework, we analyse the system sustainability using the five dimensions:

- **Individual sustainability**: Individual needs should be protected and supported with dignity and in a way that developments should improve the quality of human life and not threaten human beings;
- **Social sustainability**: Relationships of people within society should be equitable, diverse, connected and democratic;
- **Technical sustainability**: Technology must cope with changes and evolution in a fair manner, respecting natural resources;
- **Environmental sustainability**: Natural resources have to be protected from human needs and wastes; and
- **Economic sustainability**: A positive economic value and capital should be ensured and preserved.

B. *Sustainability Profiling*

Sustainability profiling has been used mostly for software energy and data centre consumption, as well as in cities and urban settlements. James [19] highlighted that a holistic and integrated understanding of urban life is essential. He presented an urban profile framework for cities sustainability including four main domains ecology, economics, politics and culture as well as seven sub-domains for each main domain. The framework was also applied to the sustainability of eLearning by Stewart and Khare [20]. This framework was providing a nine-point scale rating that is imprecise and has to be extended to fit software development process and to cover the corresponding sustainability dimensions.

Gmach et al. [21] proposed a profiling approach for the sustainability of data centres, to quantify energy during design and operation of data centres. Similarly, Jagroep et al. [22] demonstrated a software energy profiling to analyse software changes in energy consumption between releases of a software product. Although both studies focused on energy consumption that could impact environmental and economic dimensions of sustainability, individual and social dimensions were ignored in the measurement. Our approach covers the five dimensions of sustainability to quantify the sustainability of any software system, starting from the requirements phase and continuing over the phase of maintenance.

C. Quantitative Approach

Quantitative approaches are used to analyse data and to measure qualities in software engineering [23], [24]. For instance, goal-oriented requirements and user experience are analysed and measured via quantitative techniques having a rating scale of probability between satisfaction and denial of satisfaction. The rating scales and data analysis techniques vary from one quantitative approach to another. Some approaches use a five-level Likert scale while others employ a nine-point scale to present people's attitudes by scaling their responses. Notably, the Likert rating scales and the nine scales that are giving a number of options are closed format. For example, if a questionnaire has a closed five Likert scale, participants can only express their opinion through one of the five choices. These closed format options are imprecise, difficult to choose between and limited. A solution to overcome drawbacks of closed formatted scales are the fuzzy rating scale [10], cf. Section II-D for more details.

The quantitate approaches can be applied to several types of data, and the type of data to analyse might influence the choice of the approach. Tullis and Albert [23] suggest to distinguish the following four types of data:

- **Nominal data** is categorised or classification data, which it is not in any particular order, e.g., gender or hair colour;
- **Ordinal data** is ordered classified data, but the differences between them are not meaningful, e.g., product and movie ratings;
- **Interval data** is classified data where the difference between two data items is meaningful, but without natural zero points, e.g., temperature units;
- **Ratio data** is interval data with absolute zero, e.g., weight and height.

To analyse sustainability requirements, we will create from the provided by stakeholders ranking the corresponding *ratio data*. This transformation will be done using TOPSIS, cf. Section II-E. The ratio data will be then further explored to build the system profile.

D. The Fuzzy Rating Scale

A fuzzy rating scale (FRS) allows the capturing of the diversity of individual responses in questionnaires, also avoiding imprecision while rating a questionnaire [10]. For our sustainability profiling, stakeholders will be required to rate the corresponding sustainability dimensions. For example, as an alternative of stakeholders choice from a five classified rating scale, they can select their range and extend it between a range of two extreme poles.

To implement an FRS, we adopt the fuzzy rating scale method proposed by Lubiano et al. [25]:

Step 1 Considering a representative rating on the bounded interval;

Step 2 Determining a core response to be considered as *fully compatible*;

Step 3 Determining a support response to be considered as *compatible to some extent*; and

Step 4 Creating a trapezoidal fuzzy number from the two intervals, which are *linearly interpolated*, as $Tra(a, b, c, d)$, where $0 \leq a \leq b \leq c \leq d \leq 1$.

Figure 2 presents an example on application the above method to within our framework: The scale goes from 0 to 100%, where 0 corresponds to the worst case (critical value), and 100 corresponds to the best case (green value). For simplicity, it is also possible to use a scale from 0 to 1, where 1 corresponds to 100%.

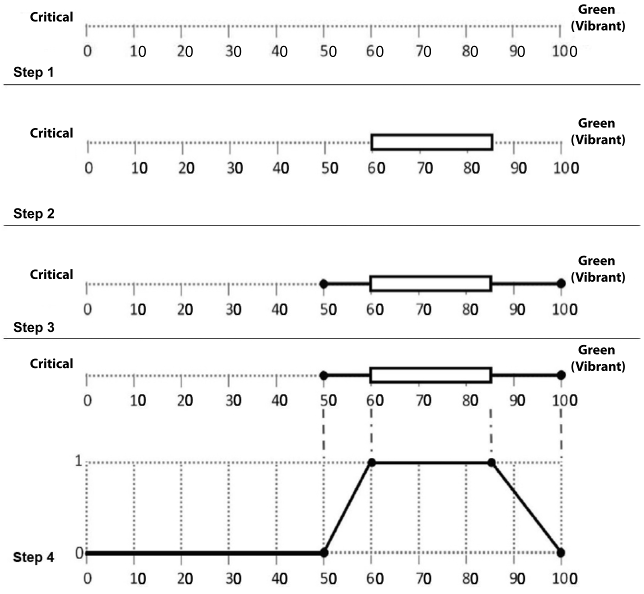

Fig. 2. Fuzzy Rating Scale for Sustainability Profiling

E. TOPSIS

Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is an effective technique to evaluate sustainability requirements which change over time is utilising. TOPSIS is one of the multiple criteria decision analysis approaches to identify the best alternative that is nearest to an ideal solution and farthest from negative ideal solution [12]. The principles of TOPSIS are simple, and positive ideal solutions and negative ideal solutions formed [26]. The benefit criteria in the positive ideal solution are maximised, and the cost criteria are minimised, while the cost criteria in the negative ideal solution are maximised, and the benefit criteria are minimised [11].

The following is the stepwise procedure of TOPSIS according to Behzadian [11]:

Step 1 Construct normalised decision matrix r_{ij}

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}, \quad for\ i = 1, \cdots, m, \quad j = 1, \cdots, n \tag{1}$$

Step 2 Construct the weighted normalised decision matrix v_{ij}

$$v_{ij} = w_i r_{ij} \tag{2}$$

where w_i is the weight for j criterion.

Step 3 Determine the positive ideal (A^*) and the negative ideal solutions ($A^{'}$):
Positive ideal solutions

$$A^* = \{max(v_{ij})|j \in J; min(v_{ij})|j \in J\prime\} = \{v_1^*, \cdots, v_n^*\} \tag{3}$$

Negative ideal solutions

$$A^{'} = \{min(v_{ij})|j \in J; max(v_{ij})|j \in J\prime\} = \{v_1^{'}, \cdots, v_n^{'}\} \tag{4}$$

Step 4 Calculate the separation measures:
The separation from positive ideal is

$$S^* = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_i^*)^2}, \quad i = \{1, \cdots, m\} \tag{5}$$

Similarly, the separation from negative ideal is

$$S^{'} = \sqrt{\sum_{j=1}^{n}(v_{ij} - v_i^{'})^2}, \quad i = \{1, \cdots, m\} \tag{6}$$

Step 5 Calculate the relative closeness to the ideal solution C_i^*

$$C_i^* = \frac{S^{'}}{(S^* + S^{'})}, \quad 0 < C_i^* < 1, \quad i = \{1, \cdots, m\} \tag{7}$$

$C_i^* = 1$ if A_i solution has the best condition,
$C_i^* = 0$ if A_i solution has the worst condition.

III. FRAMEWORK FOR SUSTAINABILITY PROFILING

The general idea of the framework workflow is presented in Figure 1. To measure the sustainability aspects of the requirements, we adopted the FRS approach. Requirements are rated against sustainability dimensions, which gives an input to the TOPSIS procedure. The provided by TOPSIS results will create a basis for sustainability profiling: using these results, our framework determines (1) the sustainable of each system requirement, (2) sustainability of the software system as whole. This will be presented in a five-star rating within each level of sustainability dimensions and the overall sustainability of each requirement. The analytical approach consists of the following five steps, cf. also Figure 1.

A. Assigning Stakeholders

Requirements engineers should assign stakeholders to one of the three stakeholder groups having end-users, administrators, and developers and providers groups. For instance, in eLearning systems the learner and instructor are in the end-users group while ITs support could be assigned to the administrator group.

B. Defining Questions

The framework will generate a questionnaire including related questions (instructions) for each requirement with regard to the sustainability dimensions and stakeholders groups. Thus, for each requirement k questions will be created, where $1 \leq k \leq 5$. Each question should present a single sustainability dimension perspective, which is covered by the requirement, and have a form
Rate the influence of the requirement on the X sustainability, where X is belongs to the set
$\{individual, social, technical, environmental, economic\}$. The generated questionnaire can be further revised and adapted by both requirements engineers and sustainability experts, before continuing with the next step.

For example, requirement R1 has to have five questions, covering each dimension of the sustainability.

C. Rating Requirements

Each stakeholder has to answer allotted question from vary views of certain sustainability dimension by using the FRS. For example, stakeholders, who are in the learners and instructors group, will answer two questions for each requirement: from the individual and from the social sustainability point of view. and another time for the social sustainability. Each answer, also, will be in a form of trapezoidal fuzzy number from the two intervals as $Tra(a, b, c, d)$, where $0 \leq a \leq b \leq c \leq d \leq 1$.

D. Analysing Sustainability Using TOPSIS

After all stakeholders answered the questionnaire, the results of the FRS approach become inputs for TOPSIS. The data will be normalised and weighted according to Equations 1 and 2, and after that the steps 3, 4 and 5 of TOPSIS need to be applied twice:

- **First round**: Applying requirements as criteria to determine overall sustainability within the separation of requirements' impacts for each requirement; and
- **Second round**: Applying sustainability dimensions as criteria to analysis each dimension within all requirements and overall sustainability rating for the software.

E. Generating Software Sustainability Profiling

The result of TOPSIS analysis including two rounds helps to generate software sustainability profiling which is visualised representing the result. The profiling includes:

- **Sustainability five-star rating** Presenting the average of $\sum C^*$ in the both rounds of sustainability dimensions and requirements;
- **Five sustainability dimensions** Illustrating each dimension level combined in pentagon or bar graph (optional) for the software having all rated requirements; and
- **Bar graph** Showing an overall sustainability for each requirement.

An example of a sustainability profile for a software system, which is created using the proposed framework, is presented in the next section (cf. Figure 5).

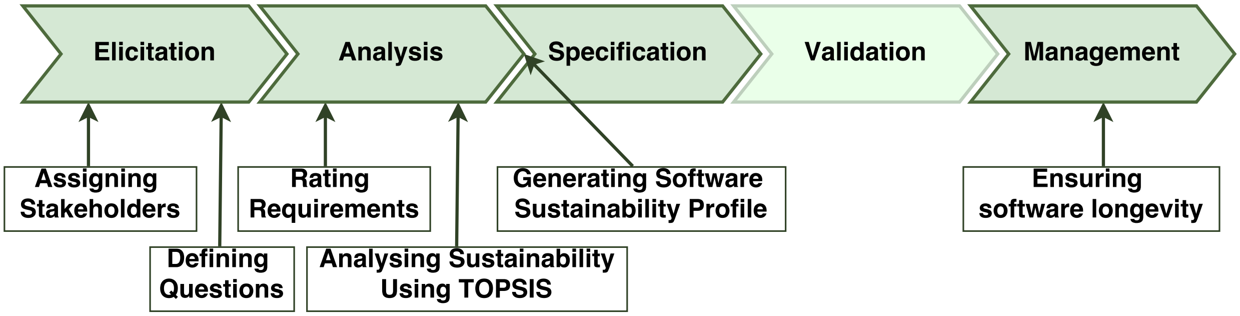

Fig. 3. Sustainability Profiling as a part of RE Activities

TABLE I
THE KEY CHART IN SOFTWARE SUSTAINABILITY PROFILING

Percentage %	Colour Code	Description
80-100	Dark green	Green (Vibrant)
60-79	Light green	Satisfactory
40-59	Yellow	Basic
20-39	Orange	Unsatisfactory
0-19	Red	Critical

Considering a different information in the profiling, we simplify and visualise the result by creating a key chart with five categories as shown in Table I. This key chart includes numeric variables in percentage, colour codes for visualisation, and linguistic variables as a description.

Figure 3 demonstrates how the proposed framework can be uses during the RE activities (we follow the definition of the RE activities introduced by [27]–[30]):

- **Requirements elicitation** is the practice of understanding and determining stakeholders' needs and constraints.
  To rate the sustainability requirements using the proposed framework, at this phase two actions are necessary:
  (A) the stakeholders have to be assigned, (B) the questionnaires have to be generated.
  However, taking into account the long-living nature of the system, re-iteration of these steps might be necessary on the management phase, to ensure the sustainability over the software system lifecycle: (A') new stakeholders can be assigned, (B') the questionnaires can be updated.
- **Requirements analysis** is the practice of refining stakeholders' needs and constraints by defining the process, data and object of the required system.
  On this phase, we conduct the following steps of our framework:
  (C) the stakeholders rate the requirements, (D) the sustainability of the system is analyses using TOPSIS, (E) the sustainability profile is generated.
  To ensure longevity of the system, these steps also can be repeated during the management phase.
- **Requirements specification** is the practice of writing down stakeholders' needs and constraints, and this documentation should be unambiguous, complete, correct, understandable, consistent, concise, and feasible.
  The sustainability profile could be seen as one of the input to the specification phase.
- **Requirements validation** is the practice of checking that the specification captures users' needs and constraints. The proposed framework does not cover the validation activities, which might be one of the future work directions.
- **Requirements management** is the practice of scheduling, controlling changes and tracking requirements over time. In the case of long-living systems, the management activities are crucial to keep the software system sustainable. The steps $(A) - (E)$ have to be repeated to provide an up-to-date sustainability profile of the system.

IV. APPLICATION OF THE PROPOSED FRAMEWORK

Let us discuss an example scenario with five requirements R1, . . . , R5. The aim of this scenario is to illustrate application of the proposed framework, without going into the technical details like generating of questions within real questionnaires. In this scenario, we will go through all framework steps and present the created sustainability profile as the final result.

A. Assigning Stakeholders

Let us assume that the requirements will be rated by ten assigned stakeholders: four in the end-users group, three in administrators group, and three in developers and providers group.

B. Defining Questions

This step is omitted in the example, as the rating activities will be simulated.

C. Rating Requirements

To simulate the rating activities where each stakeholder rates requirements against sustainability dimensions by answering defined questions, we generate random numbers between [0:1] (0 corresponds to a critical value, 1 corresponds to a green value) for the fuzzy rating scales. Figure 4 shows the results of application of the FRS approach to the requirement R1, from the prospective of ten assigned stakeholders.

As follows from Figure 4, Stakeholder S2, who is assigned to individual and social sustainability dimensions, rates R1 for

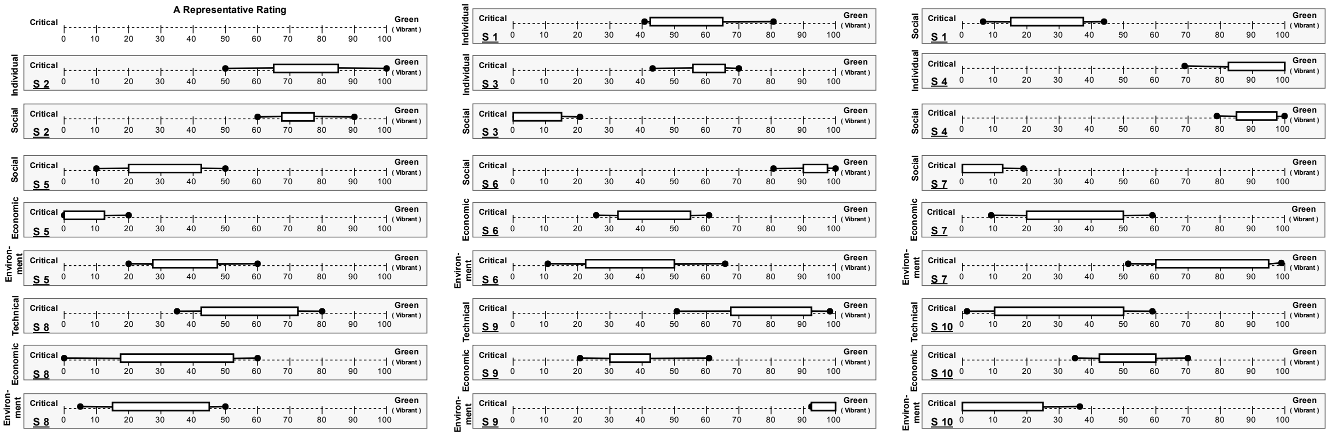

Fig. 4. Example of Fuzzy Rating Scale for Requirement (R1)

TABLE II
OUTPUT EXAMPLES OF FUZZY RATING SCALE FOR REQUIREMENTS ANALYSIS

	Dimension	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
R1	Individual	0.573	0.754	0.625	0.914						
	Social	0.276	0.727	0.087	0.917	0.377	0.942	0.066			
	Technical								0.579	0.808	0.324
	Economic					0.158	0.446	0.340	0.345	0.362	0.529
	Environment				0.382	0.351	0.799	0.291	0.986	0.130	
R2	Individual	0.281	0.472	0.232	0.289						
	Social	0.096	0.587	0.605	0.301	0.660	0.455	0.407			
	Technical								0.925	0.677	0.309
	Economic					0.093	0.506	0.738	0.567	0.459	0.395
	Environment				0.224	0.794	0.781	0.362	0.642	0.018	
R3	Individual	0.966	0.379	0.974	0.509						
	Social	0.030	0.331	0.170	0.717	0.835	0.128	0.909			
	Technical								0.173	0.157	0.728
	Economic					0.182	0.001	0.473	0.050	0.366	0.504
	Environment				0.257	0.282	0.187	0.814	0.711	0.688	
R4	Individual	0.287	0.802	0.347	0.361						
	Social	0.012	0.376	0.318	0.976	0.785	0.381	0.808			
	Technical								0.583	0.667	0.320
	Economic					0.163	0.417	0.547	0.599	0.360	0.821
	Environment				0.244	0.871	0.953	0.013	0.222	0.249	
R5	Individual	0.619	0.546	0.957	0.614						
	Social	0.600	0.005	0.460	0.003	0.977	0.535	0.518			
	Technical								0.215	0.995	0.943
	Economic					0.244	0.072	0.328	0.251	0.349	0.610
	Environment				0.214	0.704	0.662	0.949	0.714	0.583	

individual perspective as $Tra(0.51, 0.66, 0.856, 1.00)$ while social perspective as $Tra(0.60, 0.66, 0.75, 0.9)$. We calculate fuzzy values from each fuzzy rating by mean measurement, so individual and social means of R1 for S2 are 0.754 and 0.727, respectively.

D. *Analysing Sustainability*

In the next step, all the FRS outputs become inputs for TOPSIS, cf. Table II. These data are normalised according to Equation 1 for the five system requirements R1, $\ldots$, R5 within the individual, social, technical, economic and environmental dimensions of sustainability. The result of normalisation step presented in Table III.

The weighted normalisation that was constructed according to Equation 2 is showed in Table IV. Following the TOPSIS procedure, we calculate for both rounds the separation measures from positive ideal S^{*} and negative ideal solutions $S^{'}$, as well as the relative closeness C^{*}. The results are summarised in Tables V and VI. Noteworthy, we could calculate the negative impact of economic and environmental sustainability dimensions via the negative ideal solution that maximises the cost criteria and minimises the benefit criteria.

E. *Sustainability Profiling*

The generating software sustainability profiling is presented in Figure 5 within an overall of 49% sustainability which is the mean of $\sum C^{*}$ in the two rounds (in Table V and VI). Among

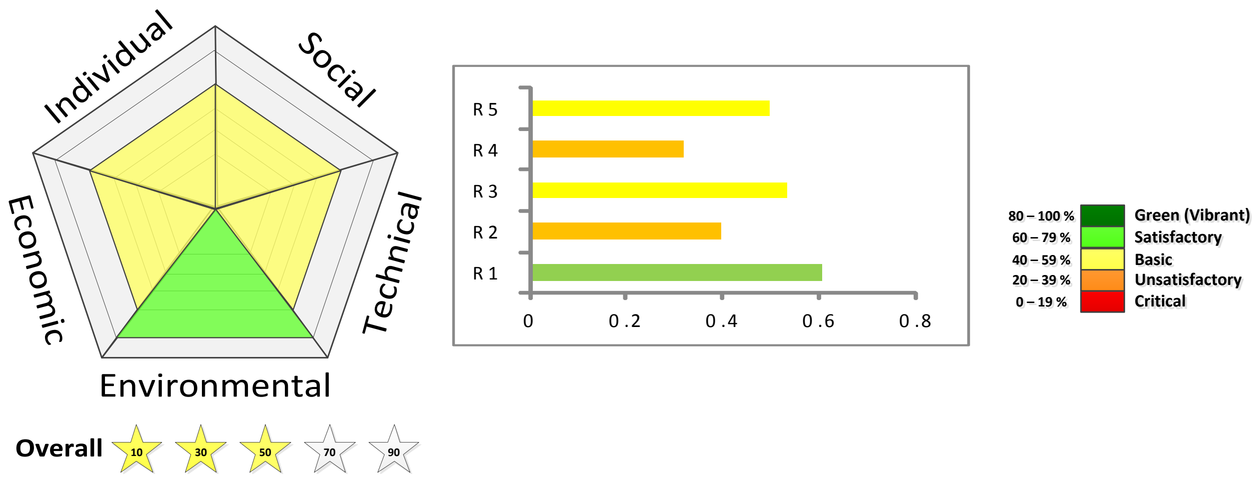

Fig. 5. Sustainability Profile of a Software System using the default colour schema. To increase accessibility of our approach, we also provide another colouring option for colour-challenged people. In this option the red colour is replaced by blue.

TABLE III
THE NORMALISATION DECISION (STEP 1) USING EQUATION 1

Dimensions	R1	R2	R3	R4	R5
Individual	0.536	0.238	0.529	0.336	0.512
Social	0.462	0.423	0.425	0.498	0.422
Technical	0.444	0.496	0.275	0.408	0.559
Economic	0.421	0.533	0.304	0.562	0.358
Environment	0.431	0.414	0.431	0.374	0.561

TABLE IV
THE WEIGHTED NORMALISATION STEPS FROM EQUATION 2

Dimensions	R1	R2	R3	R4	R5
Individual	0.146	0.033	0.166	0.063	0.126
Social	0.085	0.081	0.084	0.108	0.067
Technical	0.097	0.136	0.043	0.089	0.144
Economic	0.058	0.105	0.035	0.113	0.040
Environment	0.081	0.084	0.094	0.066	0.128

TABLE V
RESULTS OF THE STEPS 4 AND 5 IN THE FIRST ROUND

Dimensions	S*	S’	C*
Individual	0.0917	0.130	0.586
Social	0.143	0.118	0.452
Technical	0.134	0.137	0.505
Economic	0.132	0.104	0.440
Environment	0.093	0.151	0.617

TABLE VI
RESULTS OF THE STEPS 4 AND 5 IN THE SECOND ROUND

	R1	R2	R3	R4	R5
S*	0.116	0.139	0.135	0.191	0.121
S’	0.179	0.091	0.154	0.088	0.119
C*	0.607	0.394	0.533	0.317	0.497

the five requirements, R1 meets the highest level as satisfactory as well as environmental dimensions. Also, individual, social, technical and economic dimensions become basic as the lowest level of software sustainability including the five requirements in this example.

V. DISCUSSION AND CONCLUSIONS

In this paper, we introduced a framework for software sustainability profiling. We also presented and example scenario to provide a numerical illustration on how the framework can be applied. The framework allows to create the following profiling elements:

1) *Sustainability five-star rating* for overall sustainability ranking of entire software requirements;
2) *Visualisation of the five sustainability dimensions* as a pentagon graph (and, optionally, also a bar graph) for all dimension levels of the entire software requirements; and
3) *Bar graph* for overall sustainability of each requirement.

In our framework we apply a quantitative approach to measure sustainability of the software systems. The fuzzy rating scale is utilised to overcome inexplicit choices in questionnaires and increase the usability of the framework. The TOPSIS approach for requirements analysis is used to analyse ranking within the best ideal solution and the worst ideal solution among requirements that could assist to recognise the positive and negative impacts on sustainability via maximising or minimising the benefit or cost.

In the case of long-living systems, it is crucial to keep the software system sustainable over the whole lifecycle of the system. The stakeholders requirements and preferences might change over the time the system is in use, and proposed framework allows to analyse the up-to-date system sustainability profiles, based on system characteristics and the up-to-date ratings (quotations) of the corresponding requirements.

One of the core features of the framework is readability of the sustainability profiles, which also implies the usability of the proposed framework. For example, we apply the five-star rating to present sustainability ratings, as this rating is perceived as a common one in other areas: the five-star rating has become a standard for electricity consumption labelling in electronic appliances such as air conditioners and computer monitors, allowing an energy efficient choice by reducing energy use and emissions (i.e., to increase environmental sustainability).

We follow the traffic lights colouring schema, where critical values are marked red and green (vibrant) are marked green to increase readability and graphic visualisation. These colours and their descriptions have been used in *Green IT* and *Sustainability Developments*. To increase accessibility of our approach, we also provide another colouring option for colour-challenged people, where the red colour is replaced by blue. Finally, there are two options to present the five sustainability dimensions as a pentagon or bar graph because it might be argued that the pentagon graph could be harder to read and need more effort to analyse represented data than the bar graph, so we provide the bar graph option for representing the five sustainability dimensions.

Future work: The main direction of our future work on the proposed framework is to develop a tool for software sustainability profiling, allowing to perform the framework steps within a single platform. We also would like to apply the proposed framework to our earlier work on the analysis of the RE aspects of ELearning systems [31] as well as of geographically distributed systems and within global product development [32]–[34].

ACKNOWLEDGEMENT

The first author is supported by a scholarship from Umm Al-Qura University, Saudi Arabia.

REFERENCES

[1] B. Penzenstadler, A. Raturi, D. Richardson, and B. Tomlinson, "Safety, security, now sustainability: The nonfunctional requirement for the 21st century," *Software, IEEE*, vol. 31, no. 3, pp. 40–47, 2014.

[2] F. Berkhout and J. Hertin, "Impacts of information and communication technologies on environmental sustainability: Speculations and evidence," *Report to the OECD, Brighton*, vol. 21, 2001.

[3] P. Lago and T. Jansen, "Creating environmental awareness in service oriented software engineering," in *Service-Oriented Computing*. Springer, 2011, pp. 181–186.

[4] S. Naumann, M. Dick, E. Kern, and T. Johann, "The greensoft model: A reference model for green and sustainable software and its engineering," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 4, pp. 294–304, 2011.

[5] B. Penzenstadler and H. Femmer, "A generic model for sustainability with process- and product-specific instances," in *Proceedings of the 2013 Workshop on Green in/by Software Engineering*, ser. GIBSE '13. New York, NY, USA: ACM, 2013, pp. 3–8.

[6] G. G. Calienes, "Requirements prioritization framework for developing green and sustainable software using anp-based decision making," 2013.

[7] B. Penzenstadler, "Infusing green: Requirements engineering for green in and through software systems," *Christopher Arciniega, Birgit Penzenstadler TechReport UCI-ISR-14-2 June*, 2014.

[8] C. Becker, S. Betz, R. Chitchyan, L. Duboc, S. M. Easterbrook, B. Penzenstadler, N. Seyff, and C. C. Venters, "Requirements: The key to sustainability," *IEEE Software*, vol. 33, no. 1, pp. 56–65, Jan. 2016.

[9] V. G. Duffy, *Improving Sustainability through Usability*. Cham: Springer International Publishing, 2014, pp. 507–519.

[10] S. de la Rosa de Sáa, M. A. Gil, G. González-Rodríguez, M. T. López, and M. A. Lubiano, "Fuzzy rating scale-based questionnaires and their statistical analysis," *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 1, pp. 111–126, Feb 2015.

[11] M. Behzadian, S. K. Otaghsara, M. Yazdani, and J. Ignatius, "A state-of the-art survey of TOPSIS applications," *Expert Systems with Applications*, vol. 39, no. 17, pp. 13 051 – 13 069, 2012.

[12] D. Mairiza, D. Zowghi, and V. Gervasi, "Utilizing TOPSIS: A Multi Criteria Decision Analysis Technique for Non-Functional Requirements Conflicts," in *RE*, ser. Communications in Computer and Information Science, D. Zowghi and Z. Jin, Eds. Springer, 2014, vol. 432, pp. 31–44.

[13] P. Achimugu, A. Selamat, R. Ibrahim, and M. N. Mahrin, "A systematic literature review of software requirements prioritization research," *Information and Software Technology*, vol. 56, no. 6, pp. 568 – 585, 2014.

[14] E. Hull, K. Jackson, and J. Dick, *Requirements engineering*. Springer Science & Business Media, 2010.

[15] G. Lami, F. Fabbrini, and M. Fusani, *Software Sustainability from a Process-Centric Perspective*, ser. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2012, vol. 301, ch. Systems, Software and Services Process Improvement, pp. 97–108.

[16] B. Penzenstadler, *Green in Software Engineering*. Cham: Springer International Publishing, 2015, ch. From Requirements Engineering to Green Requirements Engineering, pp. 157–186.

[17] R. Goodland, "Sustainability: human, social, economic and environmental," *Encyclopedia of Global Environmental Change. John Wiley & Sons*, 2002.

[18] M. Razavian, G. Procaccianti, D. A. Tamburri *et al.*, "Four-dimensional sustainable e-services," *EnviroInfo, Sep*, 2014.

[19] P. James, *Urban sustainability in theory and practice: circles of sustainability*. Routledge, 2014.

[20] B. Stewart and A. Khare, "elearning and the sustainable campus," in *Transformative Approaches to Sustainable Development at Universities*. Springer, 2015, pp. 291–305.

[21] D. Gmach, Y. Chen, A. Shah, J. Rolia, C. Bash, T. Christian, and R. Sharma, "Profiling sustainability of data centers," in *Proceedings of the 2010 IEEE International Symposium on Sustainable Systems and Technology*, May 2010, pp. 1–6.

[22] E. A. Jagroep, J. M. van der Werf, S. Brinkkemper, G. Procaccianti, P. Lago, L. Blom, and R. van Vliet, "Software energy profiling: Comparing releases of a software product," in *Proceedings of the 38th International Conference on Software Engineering Companion*, ser. ICSE '16. New York, NY, USA: ACM, 2016, pp. 523–532.

[23] T. Tullis and B. Albert, *Measuring the User Experience*, 2nd ed., ser. Interactive Technologies. Boston: Morgan Kaufmann, 2013.

[24] J. Horkoff and E. Yu, "Analyzing goal models: Different approaches and how to choose among them," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC '11. New York, NY, USA: ACM, 2011, pp. 675–682.

[25] M. A. Lubiano, S. de la Rosa de Sa, M. Montenegro, B. Sinova, and M. ngeles Gil, "Descriptive analysis of responses to items in questionnaires. why not using a fuzzy rating scale?" *Information Sciences*, vol. 360, pp. 131 – 148, 2016.

[26] J. R. S. C. Mateo, *TOPSIS*. London: Springer London, 2012, pp. 43–48.

[27] R. H. Thayer and M. Dorfman, *Software Requirements Engineering*. Wiley-IEEE Press, 2000, ch. Introductions, Issues, and Terminology, pp. 3–40.

[28] B. Nuseibeh and S. Easterbrook, "Requirements engineering: A roadmap," in *Proceedings of the Conference on The Future of Software Engineering*, ser. ICSE '00. New York, NY, USA: ACM, 2000, pp. 35–46.

[29] P. Sawyer, I. Sommerville, and S. Viller, "Requirements process improvement through the phased introduction of good practice," *Software Process: Improvement and Practice*, vol. 3, no. 1, pp. 19–34, 1997.

[30] I. Sommerville, *Software Engineering*, 9th ed. USA: Addison-Wesley Publishing Company, 2010.

[31] A. D. Alharthi, M. Spichkova, and M. Hamilton, "Requirements engineering aspects of elearning systems," in *Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference*. ACM, 2015, pp. 132–133.

[32] M. Spichkova and H. Schmidt, "Requirements engineering aspects of a geographically distributed architecture," in *10th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2015)*, 2015.

[33] M. Spichkova, H. W. Schmidt, M. R. I. Nekvi, and N. H. Madhavji, "Structuring diverse regulatory requirements for global product development," in *Requirements Engineering and Law*. IEEE, 2015, pp. 57–60.

[34] M. Spichkova, H. Liu, and H. Schmidt, "Towards quality-oriented architecture: Integration in a global context," in *European Conference on Software Architecture Workshops*. ACM, 2015, p. 64.