Crowd-Focused Semi-Automated Requirements Engineering for Evolution Towards Sustainability

Norbert Seyff[1,2], Stefanie Betz[3], Iris Groher[4], Melanie Stade[1], Ruzanna Chitchyan[5], Letícia Duboc[6],
Birgit Penzenstadler[7], Colin Venters[8], Christoph Becker[9]
[1] University of Applied Sciences and Arts Northwestern Switzerland, Switzerland, {norbert.seyff, melanie.stade}@fhnw.ch
[2] University of Zurich, Switzerland, seyff@ifi.uzh.ch
[3] Furtwangen University, Germany, besi@hs-furtwangen.de
[4] Johannes Kepler University Linz, Austria, iris.groher@jku.at
[5] University of Bristol, UK, r.chitchyan@bristol.ac.uk
[6] La Salle-URL, Spain, l.duboc@salle.url.edu
[7] California State University Long Beach, USA, birgit.penzenstadler@csulb.edu
[8] University of Huddersfield, UK, c.venters@hud.ac.uk
[9] University of Toronto, Canada, christoph.becker@utoronto.ca

***Abstract*—Continuous requirements elicitation is an essential aspect of software product evolution to keep systems aligned with changing user needs. However, current requirements engineering approaches do not explicitly address sustainability in the evolution of systems. Reasons include a lack of awareness and a lack of shared understanding of the concept of sustainability in the RE community. Identifying and analysing the effects of requirements regarding sustainability is challenging, as these effects can have an impact on multiple stakeholders and manifest themselves in one or more sustainability dimensions at different points in time. We argue that crowd-focused semi-automated requirements engineering allows the engagement of a large number of stakeholders (including users and domain experts) in a continuous cycle of negotiation regarding the potential effects of requirements on sustainability. Based on a motivating scenario, we introduce the idea of a platform for crowd-focused requirements engineering that supports the evolution towards sustainability. For the three key aspects of this platform, we present our ongoing work and discuss early results. We outline how the platform can be utilised to improve the broader awareness and understanding of sustainability, not only for the involved crowd but also for researchers and society in general.**

***Index Terms*—Requirements Engineering, Crowd, User Feedback, Sustainability, Software Evolution.**

I. Introduction

Software systems are ubiquitous and permeate every facet of modern society [1]. To endure, software needs to be both useful and easily adaptable, as stakeholder goals, technology and environments inevitably evolve and change over time [2].

Conventionally, the impact of requirements has been considered upon a selected number of stakeholders and a limited range of concerns (such as functional expectations, cost, and a few prominent non-functional requirements such as performance and security) [3]. To the best of our knowledge, there are no established means to analyse the impact of a given requirement on sustainability. This is not surprising, as the RE community and stakeholders do not have sufficient knowledge of the concept of sustainability, including the implications of a requirement upon sustainability [4].

To address this, we propose a platform that follows a crowd-focused [5] requirements elicitation, analysis and negotiation approach. The collection and analysis of relevant data are supported with machine learning (ML) techniques to derive an understanding of what the sustainability impact of a given requirement (or software product as a collection of requirements) might be. We foresee that a discussion regarding sustainability, integrating the views of different stakeholders using this platform, will lead to a better understanding of what sustainability is for a given domain and community, and will support the adoption of sustainability in practice.

In this paper, we first outline the current state of research on sustainability and RE. We then present a motivating scenario and propose a crowd and sustainability-focused platform for (semi-)automated requirements elicitation, negotiation and analysis. This platform enables a heterogeneous and distributed group of stakeholders (the crowd) to communicate and negotiate their requirements and explore the impact of each requirement on the five sustainability dimensions. Our ongoing research already investigates three key aspects of this platform to validate this conceptual solution.

II. Understanding Sustainability and Requirements

The concept of *sustainability* can be understood in the field of software and requirements engineering as the "capacity" of a system "to endure". A closely related term, sustainable development, was defined by the Brundtland Commission as "meeting the needs of the present without compromising the ability of future generations to meet their own needs" [6]. Increasingly, it is advocated that sustainability requires simultaneous consideration of several interrelated dimensions (environmental, economic, individual, social and technical), including consideration of three orders of impact or effects of software systems [7]. Nevertheless, interdependencies between these dimensions have to be negotiated for a system under analysis [8].

There exist two perspectives, from which to tackle sustainability issues in software [9]. In *sustainability by software*, one can build software systems that directly address sustainability; a smart meter, for example, may reduce energy usage. In *sustainability in software*, one can investigate possible sustainability-related effects of software use and development. In this paper, we address both. We plan to build a platform (*sustainability by software*) for any software to become sustainable (*sustainability in software*).

As the concept of sustainability has started to be discussed in the fields of software and requirements engineering, so has the concept of *sustainability requirements*. Although there is no common definition of the term, a sustainability requirement often is seen as a non-functional requirement or software quality aligned with one or more of the sustainability dimensions [10]. Based on the fact that sustainability is a cross-cutting concern, we argue that every requirement potentially can have an impact on sustainability. This understanding has several implications for RE. Instead of being a new disjoint group of requirements, requirements having an effect on sustainability can be found within existing functional and non-functional requirements categories, and are therefore a *subgroup* of existing requirements categories. We expect that requirements positively affecting sustainability ideally have an overall long-term positive effect on one or more sustainability dimensions. This effect is expected to be significant. Furthermore, effects can manifest themselves at different points in time: They can occur immediately after the system is introduced, but also lead to accumulated effects at a later point in time. Thus, we consider it necessary to analyse and negotiate all potential effects (positive and negative) for all elicited requirements on all sustainability dimensions and orders of effects. To minimise the negative effects of such requirements, possible alternatives (requirements or design solutions) should be identified and discussed among software developers and stakeholders.

In the next section, we present a motivating scenario that highlights what this novel elicitation, analysis and negotiation of requirements could look like following a crowd-focused semi-automated approach.

III. Motivating Scenario

The city of Vienna has finished building another smart city quarter. Linda works for SustainTech, a software company that has developed several of the new services used by the smart city citizens. She is the product owner of a public transport service app that allows customers to order public, autonomous minibuses, which are shared by up to nine people for commuting within the city.

SustainTech has become a significant player in Austria because they are applying a crowd-focused semi-automated requirements engineering approach. This approach focuses on the development and evolution of sustainable software systems and is supported by a platform. Linda can still remember that she was very impressed when she first heard about this platform at a conference. At that time, most of the requirements engineers and software developers did not know about sustainability and how it relates to software development and evolution. However, things have moved on since then, due to climate change and other sustainability-related threats, and now every new software product even needs to be “sustainability certified”.

Daniel is a new user of the SustainTech public transport app. This evening, he has invited friends over for dinner and wanted to arrive home early. However, the minibus is full, and the journey takes some time. This makes Daniel think that he would like to have a priority driving service, where he alone would be using the autonomous vehicle. Daniel, of course, is aware that this priority service will cost more and that it could also be an excellent option for the service provider to earn more money. So, Daniel takes his Smartphone and uses the apps feedback mechanisms to send a message to request a priority feature. To make it simple, he takes a screenshot of the order button and adds text saying: “I want to use the minibus alone to save time.”

At this moment, Daniel did not know that SustainTech was already aware that citizens wanted to have such a priority service. The platform’s automated feedback gathering service had first identified this idea by monitoring social networks. Automated topic modelling allowed Linda and her colleagues to understand that various feedback items, shared on different social networks, belong together. Linda, therefore, has already created a new requirement about a priority service feature based on the clustered feedback. This new requirement is semi-automatically checked by the platform, which illustrates its potential effects on the sustainability dimensions using a radar graph. This graph reveals that there is a tendency that the new requirement might have a negative impact on the environmental, social and individual sustainability dimensions. To better understand this impact and the priority of the requirement, Linda prompted the platform to start a crowd-focused negotiation. Then, the platform invited the crowd to communicate further sustainability issues and priorities regarding the proposed requirement. Luckily, many affected stakeholders participated, including users, members of environmental groups, the local government, and SustainTech employees. Linda is quite happy that the platform learns about the types of concerns various stakeholders have about different dimensions of sustainability.

The platform continuously visualises the status and intermediary results of the ongoing negotiation. Linda can see that most users communicated a high priority for the feature request under discussion. The negotiation has also revealed that a priority driving service will have a substantial adverse impact on the environmental, social and individual dimensions due to an increase in solo travels. Furthermore, a more detailed process-based analysis revealed that if many citizens use the priority service, it could lead to worse average travel duration, even for those using a priority service. Finally, SustainTech has decided not to provide this service, but to look for an alternative way to speed up the public transport service with the help of the crowd.

The negotiation, analysis and decision making had happened before Daniel sent his request. The results are provided to him and other crowd members as summaries and visualisations via the integrated feedback mechanism. This helps Daniel understand that his requested feature will not be implemented due to its various negative effects on sustainability. Daniel is also informed that alternatives are being sought and he is invited to participate in finding and discussing these alternatives. He is happy that SustainTech immediately reacts to his feedback, communicates that they are still interested in his opinion, and informs him of the potential effects of such a service.

Over dinner, Daniel shares his experience with his friends: "It is remarkable! Of course, I thought that a service carrying one passenger would not be so environmentally friendly. However, I had not realised the effects it could have on our society and that it could even make my journey longer! I will watch out for the possible effects of the products and services I choose to consume in the future! What about you?".

IV. Conceptual Solution and Ongoing Work

In the motivating scenario, we have illustrated how a crowd-focused semi-automated approach that supports software evolution towards sustainability could be realised. As outlined in the scenario, this envisioned approach is based on the following three key ideas: (1) *to enable affected stakeholders (the crowd) to give feedback on a system and to negotiate this feedback including the discussion on sustainability with software companies*, (2) *to semi-automatically analyse stakeholder needs with respect to sustainability*, (3) *to support decision making and software evolution based on the results of the sustainability analysis*.

Our envisioned approach will be realised with the help of a platform that includes three key components that work together to support the continuous negotiation of stakeholder needs (see Fig. 1): (1) *CrowdFeed* allows users to communicate feedback regarding the software products and services they use and to actively participate in the negotiation, (2) *Requirements and Sustainability Service (ReSuS)* classifies, clusters, and analyses the feedback received from the *CrowdFeed* component, (3) *Requirements and Sustainability Integrator (ReS-Integrator)* supports the visualization and assessment of effects on sustainability.

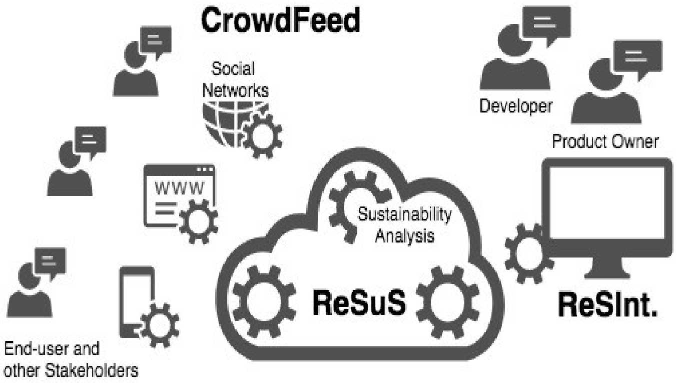

Fig. 1. Conceptual solution idea with the three key components.

The following subsections describe the three key ideas and components in more detail, outlining existing and ongoing work.

A. CrowdFeed

Link to scenario. Daniel and other stakeholders can communicate ideas, problems, and experiences on the public transport app through built-in multi-modal feedback mechanisms or social networks. This feedback can be about sustainability but is not limited to such issues, and stakeholders can communicate feedback such as usability or performance issues. The crowd is actively involved in the prioritisation of a requirement and the discussion about its sustainability impact, ultimately supporting Linda and her team in the requirement prioritisation and sustainability impact assessment.

State-of-the-art. Research on various feedback communication channels has emerged, including app stores [11] and social networks [12], but also dedicated feedback tools, designed as standalone [13], embedded [14], or cross-platform [15] solutions. The crowd can express their feedback in linguistic (e.g., free text, audio recording, category selection) or non-linguistic (e.g., rating, annotated screenshots) formats [16]. Moreover, the feedback communication can be initiated by a feedback sender like Daniel who sends (pushes) a short message or by a feedback receiver like Linda who asks (pulls) in the negotiation phase for feedback [17]. However, users may lack motivation [18], and feedback acquisition approaches might not consider users' preferences [19]. This can hinder a continuous involvement of the crowd. To the best of our knowledge, none of the existing (crowd-focused) feedback or negotiation solutions involves, supports and encourages the crowd in the elicitation and discussion of sustainability effects. The lack of such solutions hinders research investigating the crowd's interest in sustainability effects of requirements.

Link to conceptual solution. We will develop new, and modify and extend existing feedback communication channels that will enable and motivate affected stakeholders (the crowd) to give feedback to a product or service. The envisioned *CrowdFeed* component also allows stakeholders to participate in negotiation and to get informed about discussion and analysis results.

Detailed description. The *CrowdFeed* component includes push and pull feedback plug-ins that are integrated into the software system, but also mechanisms to monitor social networks. This will allow the involvement of affected stakeholders, including end-users, software engineers, and in general, any person or organisation who has an interest in the system or its sustainability effects. Moreover, *CrowdFeed* enables a feedback receiver like Linda to ask questions for clarification to a feedback sender like Daniel. *CrowdFeed* can also inform Daniel about the status of his feedback. In addition, Linda can activate a personalised rewarding system for *CrowdFeed* users to keep them motivated in the elicitation and negotiation. Overall, the *CrowdFeed* component can be configured to the feedback sender's and the feedback receiver's needs and preferences.

Ongoing work. In our ongoing work, we are investigating users' needs (e.g., privacy), preferences (e.g., feedback formats), and motives (e.g., social recognition, altruism, power) for providing feedback about a software system. From a preliminary survey of 17 participants, we have learned about several aspects that would increase their willingness to provide feedback, such as a clear statement from the feedback receiver that she is indeed interested in receiving feedback, being informed about the feedback status, or collecting points that can be exchanged for other incentives. Based on a more detailed analysis of users' needs, preferences and motives, we will design GUI variants of *CrowdFeed.* We assume that a feedback tool solution that addresses the individual needs, preferences and motives of different stakeholders increases their willingness to provide feedback and to participate in discussions.

B. ReSuS

Link to scenario. SustainTech analyses feedback messages concerning their impact on different sustainability dimensions. This analysis revealed that the requested priority transport service has a negative impact on the environmental, social and individual dimensions, encouraging SustainTech to look for alternative ways to provide efficient transportation.

State-of-the-art. ML has already been used in the area of requirements and feedback classification. For example, it has been used for automatically classifying tweets according to their relevance to different stakeholders [20]. ML can also support the classification of app reviews into categories relevant for software evolution (e.g., feature requests) [21][22]. App reviews have also been analysed by extracting requirements and sentiments [11] and filtering the most informative ones [23]. We believe that the feedback categorisation envisioned in our approach can benefit from this existing work. ML has not yet been applied to analyse feedback concerning its impact on sustainability, but our first experiments in this area (as described below) are promising.

Link to conceptual solution. The second component, the *Requirements and Sustainability Service (ReSuS)*, is the centrepiece of our conceptual solution and is composed of three subcomponents: *ReSuS-Persist*, *ReSuS-Cluster* and *ReSuS-Analyse*. *ReSuS* brings together information from different sources by continuously receiving, storing, and incrementally analysing the feedback obtained from *CrowdFeed.* It also continually communicates information to *ReSIntegrator* to support decision making.

Detailed description. The *ReSuS-Persist* sub-component is responsible for receiving and storing the feedback obtained from *CrowdFeed* in a database. The feedback is stored together with metadata and potential context information to support responses to the feedback providers. The *ReSuS-Cluster* subcomponent is responsible for feedback type categorisation (e.g., feature request, bug report) and for feedback topic clustering (e.g., priority transport service). For *ReSuS-Cluster*, ML techniques such as topic modelling [24] will be applied. Finally, the *ReSuS-Analyse* sub-component is responsible for the feedback analysis concerning the five sustainability dimensions. The goal is to identify whether a feedback cluster can potentially influence one or more sustainability dimensions. We will use ML to tag the feedback clusters with the respective influence on the sustainability dimensions (e.g., binary classification algorithms such as LinearSVC). In our example, the requested priority transport service is identified as having a negative impact on the environmental, social and individual dimensions. *ReSuS-Cluster* and *ReSuS-Analyse* will need to be customised and trained for a specific application domain (e.g., autonomous driving) and will continuously improve as the ML algorithms learn from each new feedback received by *ReSuS.* The results from the *ReSuS-Analyse* sub-component provide crucial information for decision making.

Ongoing work. As a first step, we have implemented a web-based prototype for the automatic analysis of natural language requirements concerning affected sustainability dimensions. The tool uses existing ML algorithms to classify the requirements' effects on the dimensions (multi-label classification [25]). Sustainability experts can review the suggested assessment and either approve it or perform adaptations. The algorithms continuously learn from the input provided by the experts. We have performed initial experiments with requirements of a smart home system that indicated that the automatic classification was correct in about 80% of all cases, as compared to expert classification.

C. ReSIntegrator

Link to scenario. The potential effects of a requirement on the sustainability dimensions are visualised with a radar graph to enhance Daniel's understanding. Furthermore, a more detailed sustainability assessment based on input from other stakeholder groups (e.g., domain experts) revealed that if many people use the priority transport service, a negative effect on the environmental, social and individual dimensions may occur. Finally, Daniel was informed why SustainTech could not realise his priority service idea.

State-of-the-art. For decision making only a few approaches with a focus on visualising and (manually) assessing the impact of requirements on sustainability dimensions exist. For example, Becker et al. provide a tool for visualising sustainability effects and dimensions based on a radar chart [8]. Another platform [26] enables exclusively invited stakeholders to rate the influence of requirements on sustainability dimensions; the rating will then be analysed and visualised with charts. Porras et al. [27] developed a model for manually assessing sustainability in ICT projects considering three perspectives: the dimension, level of effect, and application domain. All presented approaches provide visualisation and assist manual assessment of sustainability effects, but none support the involvement of different stakeholders.

Link to conceptual solution. The third component, *ReSIntegrator* is a tool that supports all stakeholders in understanding sustainability effects using visualisation techniques. Furthermore, these visualisations support decision makers (e.g., product owners and developers) in assessing sustainability effects. *ReSIntegrator* supports the communication between the software company, the users and other stakeholders (e.g., domain experts) via the generated visualisations, which are based on data provided by *ReSuS* and the software company's decisions regarding software product evolution.

Detailed description. We plan to develop visualisation techniques based on existing solutions [8][26][27] to show the sustainability effects of a requirement. Radar graphs can be used to show effects on various sustainability dimensions, and at the same time, BPMN process visualisations can show the change of the effects over the life cycle of a product. With the help of these visualisations, decision makers can assess the possible sustainability effects manually. This allows decision makers to consider their individual or company's priorities on the sustainability dimensions and the product lifecycle phases. We plan to provide prioritisation mechanisms building on existing approaches such as AHP [28].

ReSIntegrator supports the communication between the crowd and decision makers by providing visual information. For example, using a visualisation showing the requirements of end-user, the decision maker can click on a specific requirement, and the visualisation shows specific information based on the *ReSuS* output. For example, the number of stakeholders who think that the requirement will have a negative effect on the individual sustainability dimension. Second, *ReSIntegrator* enables the company to communicate their prioritising and their final decisions on the evolution of the software product to the stakeholders. Third, via *ReSIntegrator*, the decision makers can formulate questions to the crowd (or individual crowd members) regarding specific sustainability effects, as well as allow them to vote and comment on feedback and requirements from other stakeholders. For example, to support a software company like SustainTech in the prioritisation of requirements and the sustainability impact assessment of a specific requirement, *ReSIntegrator* can invite *CrowdFeed* users to vote, comment etc. on feedback and requirements from other stakeholders. Nevertheless, the company will have the possibility to decide what kind of information will be shown to the crowd. This means that each visualisation and decision making feature (e.g., manual assessment) can be either public or private.

Ongoing work. Our ongoing work focuses on the assessment and visualisation of sustainability effects to support users in understanding these effects and decision makers in requirements negotiation. We have developed a Domain-Specific Modeling Language for visualising and assessing Social Effects in Product Life Cycles [29]. The language visualises social sustainability risks of products. For example, there might be the risk of child work for the manufacturing of smartphones. The design is based on cognitive and conceptual principles [30] [31]. Symbols represent sustainability aspects (e.g., a lifesaver symbol means social security) and colours represent risk levels (e.g., red for a high-risk level). The language needs to be extended for our proposed conceptual solution (e.g., beyond the social dimension). Overall, it could be problematic to gather all the data needed to model sustainability because it might be not available or in the correct format.

V. Expected Impact and Conclusion

Sustainability requires one to consider multiple levels of (long-term) effects, dimensions and affected stakeholders [8]. To understand these effects, we envision a crowd-focused semi-automated requirements engineering approach to evolve existing software products towards sustainability.

The outlined platform will enable a large and diverse set of stakeholders to continuously and systematically uncover subtle and hidden concerns and to elicit, analyse, and negotiate requirements, considering their impact on all five sustainability dimensions. As this will be an iterative process, it will allow evaluating and validating these perceptions. Please note that although we focus on the discussion of sustainability issues, the described components can also be used to support the elicitation, analysis, and negotiation of user concerns regarding other issues (e.g., usability, accessibility, or performance).

Overall, we expect that our platform will lead to a better understanding and awareness of sustainability. We will learn from the crowd what sustainability means to stakeholders, and at the same time, they will learn about sustainability by interacting with our platform and discussing sustainability-related issues. For industry, we expect that our platform could assist them in assessing potential consequences of their current or planned systems. This will lead to innovative and more sustainable solutions as they are based on the insights and knowledge of affected stakeholders. We foresee that the power of the crowd, combined with automated analysis, will allow for a meaningful impact analysis of requirements on sustainability. For the research community, we expect that a better understanding of sustainability will encourage the creation of new methods and tools to foster system evolution towards sustainability. Moreover, we see our proposed platform as a research framework to investigate various topics related to crowd-focused RE, not exclusively referring to sustainability effects, such as crowd members' needs when participating in requirements elicitation and negotiation.

We are currently working on three key components that essentially will allow us to validate important aspects of the planned platform and make a proof of concept. However, it will take time, effort, and further research to develop a mature platform prototype and a market-ready product. Moreover, strategies need to be investigated not only to ensure a continuous and sufficient data stream in our platform, but also to involve other relevant sources on the development and evolution of sustainable products, such as usage data. With this paper, we want to cross disciplinary boundaries and establish a new collaboration with like-minded colleagues to reach our visionary but achievable goal.

Acknowledgment

This work was supported by the European Commission within the SUPERSEDE project (Agreement No. 644018), the NSERC RGPIN-2016-06640, and the UK EPSRC Refactoring Energy Systems fellowship (EP/R007373/1). Furthermore, the research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 712949 (TECNIOspring PLUS) and from the Agency for Business Competitiveness of the Government of Catalonia.

References

[1] R. Kitchin, and Martin Dodge, Code/Space: Software and Everyday Life, The MIT Press, 2011.

[2] C. C. Venters, R. Capilla, S. Betz, B. Penzenstadler, T. Crick, S. Crouch, E. Y. Nakagawa, C. Becker, and C. Carrillo, "Software Sustainability: Research and Practice from a Software Architecture Viewpoint," Journal of Systems and Software, vol. 138, 2018, pp. 174–188.

[3] I. Sommerville, Software Engineering, Pearson Education Limited, 10th edition, 2016.

[4] R. Chitchyan, C. Becker, S. Betz, L. Duboc, B. Penzenstadler, N. Seyff, and C. C. Venters, "Sustainability Design in Requirements Engineering: State of Practice," International Conference on Software Engineering (ICSE), 2016, pp. 533–542

[5] E. C. Groen, N. Seyff, R. Ali, F. Dalpiaz, J. Doerr, E. Guzman, M. Hosseini, J. Marco, M. Oriol, A. Perini, and M. Stade, "The Crowd in Requirements Engineering: The Landscape and Challenges," IEEE Software, vol. 34, 2017, pp. 44–52.

[6] UNWorld Commission on Environment and Development, Our Common Future, Oxford University Press, 1987.

[7] C. Becker, R. Chitchyan, L. Duboc, S. M. Easterbrook, B. Penzenstadler, N. Seyff, and C. C. Venters, "Sustainability Design and Software: The Karlskrona Manifesto," International Conference on Software Engineering (ICSE), 2015, pp. 467–476.

[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, 2016, pp. 56–65.

[9] A. Fritsch, and S. Betz, "Envisioning a Community Exemplar for Sustainability in and by ICT," International Conference on Information and Communication Technology for Sustainability (ICT4S), 2018, pp. 100–111.

[10] C. Venters, N. Seyff, C. Becker, S. Betz, R. Chitchyan, L. Duboc, D. McIntyre, and B. Penzenstadler, "Characterising Sustainability Requirements: A new species, Red Herring, or just an odd Fish?," International Conference on Software Engineering (ICSE), 2017, pp. 3–12.

[11] E. Guzman, and W. Maalej, "How do Users like this Feature? A Fine Grained Sentiment Analysis of App Reviews," International Requirements Engineering Conference (RE), 2014, pp. 153–162.

[12] N. Seyff, I. Todoran, K. Caluser, L. Singer, and M. Glinz, "Using Popular Social Network Sites to Support Requirements Elicitation, Prioritization and Negotiation," Journal of Internet Services and Applications, vol. 6, 2015, pp. 7:1–7:16.

[13] N. Seyff, G. Ollmann, and M. Bortenschlager, "AppEcho: A User-driven, in Situ Feedback Approach for Mobile Platforms and Applications," International Conference on Mobile Software Engineering and Systems (MOBILESoft), 2014, pp. 99–108.

[14] D. Dzvonyar, S. Krusche, R. Alkadhi, and B. Bruegge, "Context-aware User Feedback in Continuous Software Evolution," International Workshop on Continuous Software Evolution and Delivery (CSED), 2016, pp. 12–18.

[15] J. Hess, L. Wan, B. Ley, and V. Wulf, "In-situ Everywhere: A Qualitative Feedback Infrastructure for Cross Platform Home-IT," European Conference on Interactive TV and Video (EuroITV), 2012, pp. 75–78.

[16] I. Morales-Ramirez, A. Perini, and R. Guizzardi, "An Ontology of Online User Feedback in Software Engineering," Applied Ontology, vol. 10, 2015, pp. 297–330.

[17] W. Maalej, H.-J. Happel, and A. Rashid, "When Users become Collaborators: Towards Continuous and Context-aware User Input," Object Oriented Programming Systems Languages and Applications (OOPSLA), 2009, pp. 981–990.

[18] D. Zowghi, F. da Rimini, and M. Bano, "Problems and Challenges of User Involvement in Software Development: An Empirical Study," International Conference on Evaluation and Assessment in Software Engineering (EASE), 2015, pp. 1–10.

[19] M. Almaliki, C. Ncube, and R. Ali, "The Design of Adaptive Acquisition of Users Feedback: An Empirical Study," International Conference on Research Challenges in Information Science (RCIS), 2014, pp. 1–12.

[20] E. Guzman, R. Alkadhi, and N. Seyff, "A Needle in a Haystack: What Do Twitter Users Say about Software?," International Requirements Engineering Conference (RE), 2016, pp. 96–105.

[21] E. Guzman, M. El-Halaby, and B. Bruegge, "Ensemble Methods for App Review Classification: An Approach for Software Evolution," International Conference on Automated Software Engineering (ASE), 2015, pp. 771–776.

[22] W. Maalej, and H. Nabil, "Bug Report, Feature Request, or Simply Praise? On Automatically Classifying App Reviews," International Requirements Engineering Conference (RE), 2015, pp. 116–125.

[23] N. Chen, J. Lin, S. C. Hoi, X. Xiao, and B. Zhang, "AR-Miner: Mining Informative Reviews for Developers from Mobile App Marketplace," International Conference on Software Engineering (ICSE), 2014, pp. 767–778.

[24] T. Hofmann, "Unsupervised Learning by Probabilistic Latent Semantic Analysis," Machine Learning, vol. 42, 2001, pp. 177–196.

[25] G. Tsoumakas, and I. Katakis, "Multi-label Classification: An Overview," International Journal of Data Warehousing and Mining, vol. 3, 2007, pp. 1–13.

[26] A. D. Alharthi, M. Spichkova, and M. Hamilton, "Sustainability Profiling of Long-living Software Systems," International Workshop on Quantitative Approaches to Software Quality (QuASoQ), 2016, pp. 12–19.

[27] J. Porras, M. Palacin-Silva, O. Drögehorn, and B. Penzenstadler, "Developing a Model for Evaluation of Sustainability Perspectives and Effects in ICT Projects," International Conference on Sustainable Ecological Engineering Design for Society (SEEDS), 2017, pp. 13–14.

[28] M. Dabbagh, S. P. Lee, and R. M. Parizi, "Functional and non-functional Requirements Prioritization: Empirical Evaluation of IPA, AHP-based, and HAM-based approaches," Soft Computing 20, vol. 11, 2016, pp. 4497–4520.

[29] S. Betz, A. Fritsch, and A. Oberweis, "TracyML - A Modeling Language for Social Impacts of Product Life Cycles," ER Forum and the ER Demo Track co-located with the International Conference on Conceptual Modelling (ER), 2017, pp. 1–14.

[30] D. Moody, "The 'Physics' of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering," IEEE Transactions on Software Engineering, vol. 35, 2009, pp. 756–779.

[31] A. Fritsch, and S. Betz, "Evaluation of Social Value Icons for a Domain-Specific Modeling Language," in INFORMATIK 2017, M. Eibl and M. Gaedke, Eds. Bonn: Gesellschaft für Informatik, 2017, pp. 2323–2328.