Blueprint and Evaluation Instruments for a Course on Software Engineering for Sustainability

Birgit Penzenstadler
CSU Long Beach
Long Beach
USA
birgit.penzenstadler@csulb.edu

Stefanie Betz
Karlsruhe Insitute of Technology
Karlsruhe
Germany
stefanie.betz@kit.edu

Colin C. Venters
University of Huddersfield
Huddersfield
UK
c.venters@hud.ac.uk

Ruzanna Chitchyan
University of Bristol
Bristol
UK
r.chitchyan@bristol.ac.uk

Jari Porras
LUT
Lappeenranta
Finland
jari.porras@lut.fi

Norbert Seyff
FHNW
Windisch
Switzerland
norbert.seyff@fhnw.ch

Leticia Duboc
La Salle
Barcelona
Spain
lduboc@salleurl.edu

Christoph Becker
University of Toronto
Toronto
Canada
christoph.becker@utoronto.ca

September 11, 2018

Abstract

We report on a summer school course on Software Engineering for Sustainability (SE4S). We provide a detailed blueprint of the contents taught and its evaluation with the instruments that were used.

1 Intro and Context

Sustainability has become an important concern across many disciplines, and software systems play an increasingly central role in addressing it. However, teaching students from software engineering and related disciplines to effectively act in this space requires interdisciplinary courses that combines the concept of sustainability with software engineering practice and principles. Yet, presently little guidance exist on which subjects and materials to cover in such courses and how, combined with a lack of reusable learning objects.

Penzenstadler et al. [14] describe a summer school course on Software Engineering for Sustainability (SE4S). In the paper at hand, we provide a detailed blueprint for this course, in the hope that it can help the community develop a shared approach and methods to teaching SE4S. The course blueprint, availability of used materials and report of the study results make this course viable for replication and further improvement.

Furthermore, the paper at hand presents in detail the instruments that were used for the evaluation of the course. We used three evaluation instruments:

1. a pre-survey,
2. a post-survey, and
3. a list of evaluation criteria for the analysis of the artifacts produced by the students in the course.

2 The SE4S Course

The week-long course is designed as part of a summer school held at the Lappeenranta University of Technology (LUT) in Finland.

The learning objectives and their assessment are as follows:

1. **Sustainability concepts and principles**: Develop an understanding of the concept of sustainability and its different dimensions and orders of effect and an ability to transfer these concepts to other application domains. *Assessment*: Students will demonstrate their mastery of sustainability concepts in demonstrating the transfer to a different application domain in their team project documentation.
2. **Requirements Engineering**: Students develop an understanding of the basics of requirements engineering, they understand and are able to apply stakeholder modeling, goal modeling, process modeling, use case modeling,

and SysML. *Assessment*: Students demonstrate their mastery of requirements engineering by developing a consistent specification that includes stakeholders, goals, process model, use cases, and SysML diagrams.

3. **Systems thinking**: An understanding of the mindset of and the general principles of systems thinking, including holistic viewpoints and iterative development. Understand and be able to reason about long-term effects that a system under development may have on the environment and on society. *Assessment*: Students demonstrate their knowledge in taking a bigger picture perspective and holistic viewpoint in their rich picture. Demonstrate the reasoning in providing an analysis to that regard for the system under development and pointing out risks that may or may likely occur in the future given certain conditions.

4. **Design thinking**: Understand and be able to apply design thinking on (complex) problems, which requires alternating between narrowing down and opening up the perspective. *Assessment*: Demonstrate the application of design thinking on a local problem to demonstrate understanding and transfer of the concepts of iterative development in innovation in their project.

The subject areas and modules are detailed in Tab. 1. For each, we provide content, example key references, and rationale for their inclusion in the course. In the following, we add more details on the content of each lecture module:

1. **Sustainability Foundations**: This module explains the fundamental concepts for talking about, analyzing, and designing for sustainability. This includes the topics of Dimensions of sustainability, orders of effect, application domains, and the Sustainable Development Goals. Recommended readings are [6, 2, 5]. The module provides a common basis for scoping sustainability within this course and puts the concept into a larger perspective.

2. **Principles of sustainability design**: This module delivers and explains the principles of substitution, decoupling, and dematerialization (see Hilty and Aebischer [6]) as well as Software Engineering for Sustainability examples [11, 3].

3. **Rich pictures**: This module introduces the technique of rich pictures, a method for scoping and high-level domain modeling, as proposed by Monk and Howard [9]. Rich pictures are a simple, non-technical method to illustrate the vision for a system or scoping of a problem in its surrounding application domain and operational context. The connection to sustainability lies in the fact that rich pictures provide a more coarse-grained picture of the larger context in which a system resides, and makes the relationships between stakeholders and problem domain clear.

4. **Stakeholders and goal models**: The stakeholders that for the first time show up in the rich picture are now analyzed in more detail. Based on the reference model of Penzenstadler et al. [12], the stakeholder model makes sure all relevant roles are included. The goal modeling is based on the generic sustainability goal reference model by Penzenstadler and Femmer [13] and provides a basis for consensus, conflict identification and trade-offs with specific focus on the different aspects of sustainability. The module is an introduction to concepts, roles, reference models, analysis, and notations for both of the models.

5. **Process modeling**: This modules gives an introduction to concepts of and notation for business process modeling [10, 8] and thereby provides the transition from high-level goals to the operationalization of these objectives in executable processes. This is the important tie-in of sustainability goals to the implementation - the step needs to be made traceable for later assessment.

6. **Software engineering**: We provide an overview of the general Software Engineering process phases and introduction to use cases [7], such that usage behavior can now be tied in with sustainable human computer interface design. This complements the technology-agnostic processes with the technology-aware perspective by describing the interaction between user and system.

7. **SysML**: The introduction to SysML provides the foundation for analyzing and designing complex systems according to [4]. The Systems Modeling Language is a widely used general purpose language for modeling and verifying software systems and its widespread usage across industry makes it a good exemplar for this part of the course.

8. **Design Thinking**: The d.school's [15] highly interactive workshop on design thinking facilitates transition into rapid prototyping. It's a hands-on crash course tutorial in design thinking that has students team up and work on physical prototypes. User engaging with physical prototypes has brought about many sustainability concerns in human computer interaction design and proved useful to get literally "a better feeling" for the products and services under development.

9. **Sustainability Analysis**: The overview of all impacts of a system from short-term to long-term enables software engineers to take a wider perspective and better judge the consequences of their choices during development. We introduce the analysis and estimation of a system's impact according to the sustainability dimensions and order of effect as described in [1]. This simple tool with a one-page template enables students to perform a high-level long-term assessment of a system under development within half an hour. While the assessment outcomes of such a short time should not be used as decision basis, they point to the most important

questions and potential consequences of the long-term use of a system that should be investigated.

Table 1: Subject areas and modules

Module	***Content***	***Key refs.***	***Rationale for inclusion***
Sustainability foundations	Dimensions of sustainability, orders of effect, application domains, Sustainable Development Goals	[6, 2, 5]	Provides a common basis for scoping sustainability within this course and puts the concept into a larger perspective.
Principles of sustainability design	Principles of substitution, decoupling, and dematerialization; Software Engineering for Sustainability examples	[6, 11, 3]	Introduces the principles that can be used for thinking of system ideas for the projects to be developed during the course.
Rich pictures	Rich picture method for scoping and high-level domain modeling	[9]	Rich pictures are a simple, non-technical method to illustrate the vision for a system or scoping of a problem in its surrounding application domain and operational context.
Stakeholder and goal models	Introduction to concepts, roles, reference models, analysis, and notations for both of the models.	[12, 13]	Forms the basis for eliciting requirements for a chosen project idea. The stakeholder model makes sure all relevant roles are included, the goal model provides a basis for consensus, conflict identification and trade-offs.
Process modeling	Introduction to concepts of and notation for business process modeling	[10, 8]	Provides the transition from high-level goals to the operationalization of these objectives in executable processes.
Software Engineering	Overview of the general Software Engineering process phases and introduction to use cases	[7]	Complements the technology-agnostic processes with the technology-aware perspective by describing the interaction between user and system.
SysML	Introduction to SysML for analyzing and designing complex systems	[4]	The Systems Modeling Language is a widely used general purpose language for modeling and verifying software systems.
Design Thinking	Hands-on crash course tutorial in design thinking	[15]	The d.school's highly interactive workshop on design thinking facilitates transition into rapid prototyping.
Sustainability analysis	Introduction to the analysis and estimation of a system's impact according to the sustainability dimensions and order of effect	[1]	The overview of all impacts of a system from short-term to long-term enables software engineers to take a wider perspective and better judge the consequences of their choices during development.

The planned schedule of modules of the course and the how we modified it according to circumstances are depicted in Fig. 1.

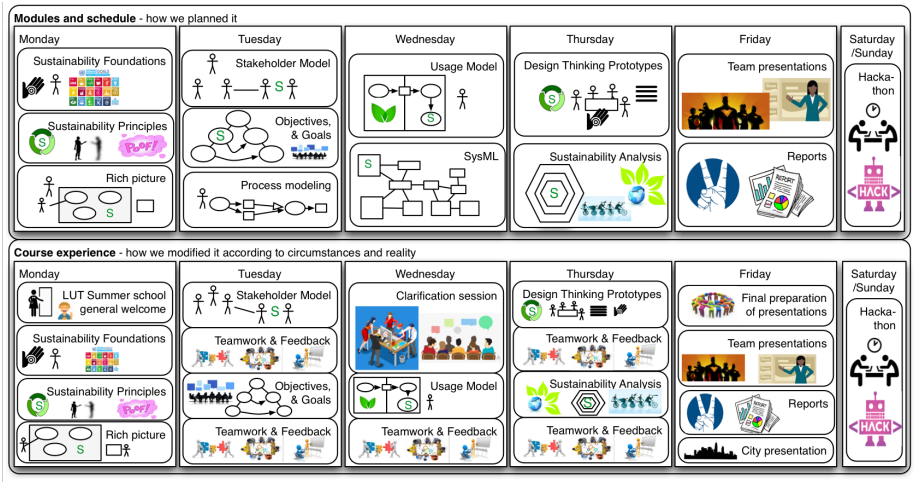

Figure 1: SE4S course planned schedule and actual experience

3 Evaluation Instruments

We used a pre- and post-survey and a report for the students' self-assessment and learning perceptions, and for the external assessment in form of artifact analysis we used a criteria catalogue.

3.1 Pre-Survey before the SE4S course

A short pre-survey (Table 2) evaluates whether our (informal and internal) hypothesis about the characteristics of the student population would hold, and to be able to better tailor the course to the student population that decided to register for the course in that instance.

The survey contained questions on their familiarity and experience with the general software engineering process, requirements elicitation and modeling, UML diagrams, SysML, IFML (Interaction Flow Modelling Language), Attribute-Driven Design (ADD), user interface development, rapid prototyping, design thinking, systems thinking, and computational thinking.

Table 2: Pre-Survey

1. Do you know the general software engineering process (the phases, roles)?
2. Have you taken a course on software engineering?
3. Have you ever developed a software project (by yourself or in a team)? Please briefly describe what you did and which technologies you used (bullet points are fine).
4. Have you written down requirements, e.g. in natural language?
5. Have you ever modeled requirements, e.g. in use cases?
6. Have you used UML diagrams for software design?
7. Have you used sysML for systems design?
8. Have you used IFML (Interaction Flow Modelling Language) for software design?
9. Have you used Attribute-Driven Design (ADD) for software architecture design?
10. Have you ever developed a user interface / graphical interface?
11. Have you used rapid prototyping?
12. Have you heard of "design thinking"? If so, have you ever used it?
13. Have you heard of systems thinking? If so, have you ever applied it?
14. Have you heard of computational thinking?
15. Have you already signed up for the hackathon on the weekend August 5/6?

3.2 Post-Survey after the SE4S course

We used a survey (Table 3) that consisted of several sections, one part dedicated to the SE4S course, one part on ethics perceptions, and one part on value ratings.[1] The part specific for the SE4S course was composed by a background section, the motivation for the course selection, a section on their notions of sustainability, software and software engineering, their perceptions of the course content, and reflections on the course.

Table 3: Post-Survey

Background
2. Gender
3. What is your age range?
4. What is your highest education level?
5. What is your highest qualification?
6. What is your discipline?
Motivation for Course Selection
7. Why did you choose this course?
8. What did you expect to learn from this course?
Notions of Sustainability, Software and Software Engineering
9. What did you think sustainability is before you started this course?
10. What do you think sustainability is now, after completing the course?
11. How much has your perception of sustainability changed as result of this course?
12. Before starting this course, did you think software could help with sustainability? If yes, then how?
13. After completing this course, how has your perception changed: do you think software can help with sustainability? And If yes: how?
14. Before starting this course how much did you think can software help with achieving sustainability?
15. After completing this course, how much, do you think, software can help with achieving sustainability?
16. What element of the course changed your perception, if any?
17. What did you think software engineering was before you started this course?
18. What do you think software engineering is now, after completing the course?
19. Before starting this course, how did you think software *engineering* could help with sustainability?
20. After completing this course, how do you think software *engineering* can help with sustainability?
21. Before starting this course how much, did you think, can software *engineering* help with achieving sustainability?
22. After completing this course, how much, do you think, software *engineering* can help with achieving sustainability?

[1] Penzenstadler et al. [14] analyzes the parts of the survey dedicated to the SE4S course due to space limitations.

23. What did you think "software engineering for sustainability" was before completing this course? 24. What do you think "software engineering for sustainability" is now, after completing the course? 25. What helped you most to link the notions of software engineering and sustainability? (You can consider the material taught in this module as well as any other sources).
On Course Content 26. What three key things related to sustainability did you learn in this course? 27. What three key things related to software engineering did you learn from this course? 28. How useful do you think the following techniques are for Software Engineering for Sustainability? (Rating rich picture, stakeholder modeling, goal modeling, use case modeling, design thinking and sustainability analysis on a Likert scale (1: not at all - 5: all the time)) 29. Which of the following techniques would you apply in your further work/research? (rating of same techniques on same Likert scale)
Reflections 30. What would you like to do next with the knowledge you learned in this course? 31. What would you like to learn next about sustainability and/or software engineering? 32. What did you especially like about this course? 33. What did you especially dislike about this course?

3.3 Criteria for the analysis of the artefacts

Over the course of the week, the objective was to develop a specification according to a small requirements artifact model as well as some prototypes or mock-ups. An overview of the artifact model to be produced is given in Fig. 2. The artifacts are a rich picture, a stakeholder model, a goal model, a use case overview model, design thinking prototypes, and a sustainability analysis diagram. For the artifact analysis, we used a list of jointly elaborated quality criteria to structure their analysis (Table 4). For each artifact, there is a number of questions and criteria to be considered by the evaluators.

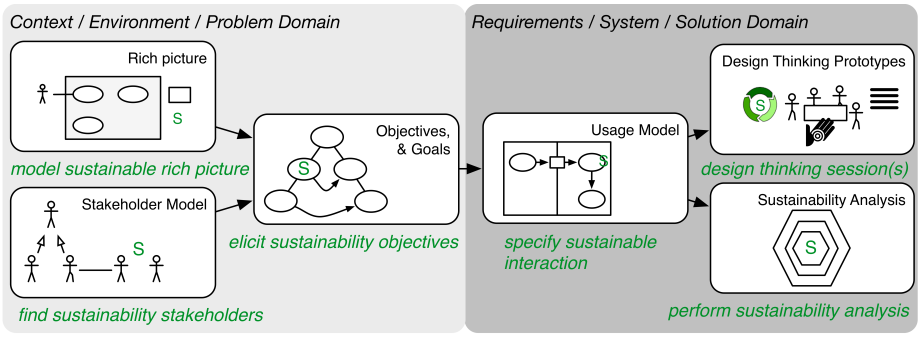

Figure 2: RE4S artefact model used in the course

Table 4: Criteria for the analysis of the artefacts

Overall • Does the project address the chosen sustainability challenge well?
Stakeholder model • Have all major stakeholder groups been considered? • Is the diagram easy to understand? • Are there well-organized hierarchies? • Are the relationships between the stakeholders clear? • Have stakeholders been analyzed and described (to an adequate degree of detail)?

Goal model

- Is the goal model well structured?
- Are they broken down and refined into sub-goals where possible?
- Are the influences and conflicts between sustainability and other goals represented?
- Are all stakeholders? main concerns represented by goals?
- Is a good accompanying explanation provided for the diagram?

Rich picture

- Can I (picturing myself as a stakeholder) easily understand what the problem is about?
- Is the vision described and illustrated to an adequate degree of detail?

Usage model

- Is a use case overview diagram provided and includes all important use cases?
- Does it have a system boundary, an actor outside that boundary, and relations from the actor to all use cases?

Sustainability analysis

- Are all dimensions analyzed?
- Are all orders of effect analyzed?
- Are all orders of effect analyzed for all dimensions?
- Are cross-relations denoted between the effects? Does every second order effect have a first order effect it expands on? Does every third order effect have a second order effect it expands on?
- Do the predictions in the individual fields describe actually possible effects of the system on its environment (in the opinion of the reviewer)?

4 Conclusion

This paper provides a detailed blueprint for a course on software engineering for sustainability. The authors are planning to replicate this course every year in different locations and report on the continued evaluation. We hope the blueprint and evaluation instruments serve further researchers and educators in developing their own courses on software engineering for sustainability.

References

[1] Christoph Becker, Stefanie Betz, Ruzanna Chitchyan, Leticia Duboc, Steve M Easterbrook, Birgit Penzenstadler, Norbet Seyff, and Colin C Venters. Requirements: The key to sustainability. *IEEE Software*, 33(1):56–65, 2016.

[2] Christoph Becker, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Birgit Penzenstadler, Norbert Seyff, and Colin C Venters. Sustainability design and software: The karlskrona manifesto. In *ICSE-SEIS*, volume 2, pages 467–476. IEEE, 2015.

[3] Ruzanna Chitchyan et al. Evidencing sustainability design through examples. In *4th Intl. Workshop RE4SuSy*, 2015.

[4] Sanford Friedenthal, Alan Moore, and Rick Steiner. Omg systems modeling language (omg sysml) tutorial. In *INCOSE Int. Symposium*, volume 18, pages 1731–1862, 2008.

[5] David Griggs et al. Policy: Sustainable development goals for people and planet. *Nature*, 495(7441):305–307, 2013.

[6] Lorenz M Hilty and Bernard Aebischer. Ict for sustainability: An emerging research field. In *ICT Innovations for Sustainability*, pages 3–36. Springer, 2015.

[7] Gerald Kotonya and Ian Sommerville. *Requirements engineering: processes and techniques*. Wiley Publishing, 1998.

[8] Selim Larsch, Stefanie Betz, Leticia Duboc, Andréa Magalhães Magdaleno, and Camilla Bomfim. Integrating sustainability aspects in business process management. In *Business Process Management Workshops*, pages 403–415, 2016.

[9] Andrew Monk and Steve Howard. Methods & tools: the rich picture: a tool for reasoning about work context. *interactions*, 5(2):21–30, 1998.

[10] Martyn A Ould and MA Ould. *Business Processes: Modelling and analysis for re-engineering and improvement*, volume 598. Wiley Chichester, 1995.

[11] B. Penzenstadler. Infusing green: Requirements engineering for green in and through software systems. In *Third Intl. Workshop RE4SuSy*, 2014.

[12] B. Penzenstadler et al. Who is the advocate? stakeholders for sustainability. In *Proc. of the 2nd Intl. Workshop on Green and Sustainable Software*, pages 70–77. IEEE, 2013.

[13] B. Penzenstadler and H. Femmer. A generic model for sustainability with process-and product-specific instances. In *Workshop on Green in/by software engineering*, pages 3–8. ACM, 2013.

[14] Birgit Penzenstadler, Stefanie Betz, Colin C. Venters, Ruzanna Chitchyan, Jari Porras, Norbert Seyff, Leticia Duboc, and Christoph Becker. Everything is INTERRELATED: Teaching Software Engineering for Sustainability. In *International Conference on Software Engineering: Track on Software Engineering Education and Training*, 2018.

[15] Stanford d.school. A virtual crash course in design thinking, 2017. last accessed Oct 21 2017.