

SYLLABUS

SUSTAINABLE SOFTWARE ENGINEERING

University year 2025-2026

1. Information regarding the programme

1.1. Higher education institution	Babeş-Bolyai University
1.2. Faculty	Mathematics and Computer Science
1.3. Department	Computer Science
1.4. Field of study	Computer Science
1.5. Study cycle	Master
1.6. Study programme/Qualification	Software Engineering
1.7. Form of education	Full time

2. Information regarding the discipline

2.1. Name of the discipline		Sustainable Software Engineering					Discipline code		MME8192
2.2. Course coordinator					Lect. phd. Vladiela Petraşcu				
2.3. Seminar coordinator					Lect. phd. Vladiela Petraşcu				
2.4. Year of study	1	2.5. Semester	1	2.6. Type of evaluation	E	2.7. Discipline regime		Mandatory	

3. Total estimated time (hours/semester of didactic activities)

3.1. Hours per week	4	of which: 3.2 course	2	3.3 seminar/laboratory/project	2
3.4. Total hours in the curriculum	56	of which: 3.5 course	28	3.6 seminar/laboratory/project	28
Time allotment for individual study (ID) and self-study activities (SA)					hours
Learning using manual, course support, bibliography, course notes (SA)					28
Additional documentation (in libraries, on electronic platforms, field documentation)					20
Preparation for seminars/labs, homework, papers, portfolios and essays					28
Tutorship					14
Evaluations					4
Other activities:					
3.7. Total individual study hours	94				
3.8. Total hours per semester	150				
3.9. Number of ECTS credits	6				

4. Prerequisites (if necessary)

4.1. curriculum	Software Engineering lecture
4.2. competencies	

5. Conditions (if necessary)

5.1. for the course	Videoprojector
5.2. for the seminar /lab activities	Videoprojector, computers, Internet access

6.1. Specific competencies acquired ¹

Professional/essential Competencies	• understanding and working with basic concepts in software engineering; • modeling and solving real-life problems;
Transversal competencies	• team work capabilities; able to fulfill different roles; • professional communication skills; concise and precise description, both oral and written, of professional results , negotiation abilities;

6.2. Learning outcomes

Knowledge	• The student proves knowledge related to specifying the requirements of research activities in the domain of computer science in general and software engineering in particular and he/she understands the role of research in promoting progress • The student possesses the fundamental knowledge for modelling, being able to analyse real life problems and to translate them in concrete requirements and to design a corresponding software model
Skills	• The student can apply advanced information system knowledge starting from a high level of abstraction and being able to offer implementation solutions for complex software system • The student has the ability to combine information in different ways in order to form a positive attitude towards his/her own development
Responsibility and autonomy:	• The student proves abilities to work independently in order to obtain knowledge necessary for designing, managing and evaluating research activities in software engineering • The student uses efficient strategies, methods and techniques for lifelong education, in order to self-educate and self-develop his/her personal and professional skills

7. Objectives of the discipline (outcome of the acquired competencies)

7.1 General objective of the discipline	• Make students acknowledge that any software system, as part of a digitalization project, has various types of impacts on several sustainability dimensions and guide them towards learning how to assess the sustainability of a software solution and to reason about tradeoffs among different sustainability quality concerns.
---	---

¹ One can choose either competences or learning outcomes, or both. If only one option is chosen, the row related to the other option will be deleted, and the kept one will be numbered 6.

7.2 Specific objective of the discipline	At the end of the course, students will: • have basic knowledge regarding software sustainability and define it as a complex software quality attribute, encompassing various dimensions and types of impact; • have knowledge of software sustainability principles, sustainability reference models and associated tools; • be able to reason about sustainability-related concerns, assess the trade-offs between different such concerns and apply that reasoning to a concrete project, with the aid of a specific toolset; • be able to reflect critically on the issues learned, their impact on society, and express own motivated opinions.
---	--

8. Content

8.1 Course	Teaching methods	Remarks
C1. Introduction to Software Sustainability	Description	
C2. Software Sustainability Priciples. The Karlskrona Manifesto	Explanation	
	Conversation	
	Debate	
	Case studies	
C3. The Karlskrona Manifesto: Case Studies	Flipped classroom	
	Case studies	
C4. Software Sustainability Reference Models	Description	
C5. Software Sustainability Awarness Framework (SuSAF)	Explanation	
C6. Software Sustainability Assessment Framework (SAF)	Conversation	
	Debate	
	Case studies	
C7. SuSAF & SAF: Case Studies	Flipped classroom	
	Case studies	
C8. Invited Lecture	Description	
C9. Sustainable Software Architecture. Technical Debt		
C10. Architectural Styles that Reduce Technical Debt		
C11. Environmental Impact of Software. Digital Carbon Footprint		
C12. Developing Energy Efficient Software (I)		
C13. Developing Energy Efficient Software (II)		
C14 Green Engineering at Scale		
Bibliography		
[1] Becker, Christoph. (2014). <i>Sustainability and longevity: Two sides of the same quality?</i>. Mental. 20. [2] Becker, Christoph & Chitchyan, Ruzanna & Duboc, Leticia & Easterbrook, Steve & Penzenstadler, Birgit & Seyff, Norbert & Venters, Colin. (2015). <i>Sustainability Design and Software: The Karlskrona Manifesto</i>. 467-476. 10.1109/ICSE.2015.179. [3] Calero, C., Moraga, M. Á., & Piattini, M. (Eds.). (2021). <i>Software sustainability</i>. Springer Cham. [4] Condori-Fernández, Nelly & Lago, Patricia. (2017). <i>Characterizing the Contribution of Quality Requirements to Software Sustainability</i>. Journal of Systems and Software. 137. 10.1016/j.jss.2017.12.005. [5] Currie, A., Hsu, S., & Bergman, S. (2024). <i>Building green software: A sustainable approach to software development and operations</i>. O'Reilly Media. [6] Fatima, Iffat & Lago, Patricia. (2024). <i>Software Architecture Assessment for Sustainability: A Case Study</i>. 10.1007/978-3-031-70797-1_16. [7] Fontanarrosa, S. (2024). <i>Green software engineering: Exploring green technology for sustainable IT solutions</i>. Packt Publishing. [8] Lago, Patricia & Akinli Kocak, Sedef & Crnkovic, Ivica & Penzenstadler, Birgit. (2015). <i>Framing Sustainability as a Property of Software Quality</i>. Communications of the ACM. 58. 70-78. 10.1145/2714560. [9] Lago, P., Fernandez, N. C., Fatima, I., Funke, M., & Malavolta, I. (2024). <i>The Sustainability Assessment Framework Toolkit: A Decade of Modeling Experience</i>. arXiv preprint arXiv:2405.01391. [10] Lago, Patricia & Penzenstadler, Birgit. (2017). Editorial: <i>Reality check for software engineering for sustainability-pragmatism required</i>: Editorial. Journal of Software: Evolution and Process. 29. e1856. 10.1002/smr.1856.		

- [11] Lago, Patricia & Verdecchia, Roberto & Condori-Fernández, Nelly & Rahmadian, Eko & Sturm, Janina & Nijnanten, Thijmen & Bosma, Rex & Debuysscher, Christophe & Ricardo, Paulo. (2021). *Designing for Sustainability: Lessons Learned from Four Industrial Projects*. 10.1007/978-3-030-61969-5_1.
- [12] Lilienthal, C. (2019). *Sustainable software architecture: Analyze and reduce technical debt*. Dpunkt Verlag.
- [13] Naumann, Stefan & Dick, Markus & Kern, Eva & Johann, Timo. (2011). *The GREENSOFT Model: A reference model for green and sustainable software and its engineering*. Surface Science - SURFACE SCI. 1. 10.1016/j.suscom.2011.06.004.
- [14] Noman, Hira & Mahoto, Naeem & Bhatti, Sania & Abosaq, Hamad & Al Reshan, Mana & Shaikh, Asadullah. (2022). *An Exploratory Study of Software Sustainability at Early Stages of Software Development*. Sustainability. 14. 8596. 10.3390/su14148596.
- [15] Sriraman G, Raghunathan S. (2023). *A Systems Thinking Approach to Improve Sustainability in Software Engineering—A Grounded Capability Maturity Framework*. Sustainability. 15(11):8766. <https://doi.org/10.3390/su15118766>
- [16] Oyedeji S, Seffah A, Penzenstadler B. (2018). *A Catalogue Supporting Software Sustainability Design*. Sustainability. 10(7):2296.
- [17] Oyedeji, S., Penzenstadler, B. (2020). *Experiences from Applying the Karlskrona Manifesto Principles for Sustainability in Software System Design*. Proceedings of the 8th International Workshop on Requirements Engineering for Sustainable Systems.
- [18] Venters, Colin & Jay, Caroline & Lau, Lydia & Griffiths, Michael & Holmes, Violeta & Ward, Rupert & Austin, Jim & Dibsedale, C.E. & Xu, J.. (2014). *Software sustainability: The modern tower of babel*. CEUR Workshop Proceedings. 1216. 7-12.

Links

[SAF] [SAF-Toolkit homepage](#)

[SusAF] [Sustainability Awareness Framework \(SusAF\)](#)

8.2 Seminar / laboratory	Teaching methods	Remarks
S1: Introduction to software sustainability Seminar activities management	Description Dialog Explanation	Two seminar hours every other week
S2 -> S6: Exploring software sustainability in scientific papers	Mini-workshop Student-led presentation Guided debate Peer/instructor feedback Reflection and self-assessment	
S7: Sustainability analysis project discussions	Conversation Explanation Case studies	

9. Corroborating the content of the discipline with the expectations of the epistemic community, professional associations and representative employers within the field of the program

- This course follows the IEEE and ACM Curricula Recommendations for Software Engineering studies.
- Courses with similar content are taught in the major universities offering similar study programs.
- Course content is considered important by the software companies, for improving the reliability of the resulting software products.

10. Evaluation

Activity type	10.1 Evaluation criteria	10.2 Evaluation methods	10.3 Percentage of final grade
10.4 Course	• Understanding of core concepts • Analytical and critical thinking • Project work • Communication skills	Exam session Team assignment – Sustainability analysis project (participate in small teams to a project focused on assessing the sustainability of a chosen digital solution)	70%

	• Understanding of core concepts • Analytical and critical thinking	Re-take session Written exam	70%
10.5 Seminar/laboratory	• Paper understanding • Critical analysis • Organisation and clarity • Communication and delivery	Individual seminar assignment - <i>Scientific paper presentation</i> (prepare a talk and a short report on a software sustainability-related scientific paper of one's choice)	30%
10.6 Minimum standard of performance			
• Average grade 5			

11. Labels ODD (Sustainable Development Goals)²

Not applicable.

Date:

04.11.2025

Signature of course coordinator

Lect. phd. Vladiela PETRAȘCU

Signature of seminar coordinator

Lect. phd. Vladiela PETRAȘCU

Date of approval:

Signature of the head of department

Assoc. prof. phd. Adrian STERCA

² Keep only the labels that, according to the [Procedure for applying ODD labels in the academic process](#), suit the discipline and delete the others, including the general one for *Sustainable Development* – if not applicable. If no label describes the discipline, delete them all and write „*Not applicable.*”.