POLI ESCOLA SUPERIOR TECNOLOGIA GESTÃO TÉCNICO GUARDA	**SUBJECT DESCRIPTION**	**MODELO** PED.013.03

Course	**Computer Science**					
Subject	**Software Engineering II**					
Academic year	2023-2024	*Curricular year*	3rd	*Study period*	1st semester	
Type of subject	Compulsory	*Student workload (H)*	Total: 168	Contact: 75	*ECTS*	6
Professor(s)	Maria Clara dos Santos Pinto Silveira (PhD)					
☒ *Area/Group Coordinator* ☐ *Head of Department* *(select)*	Maria Clara dos Santos Pinto Silveira (PhD)					

PLANNED SUBJECT DESCRIPTION

1. LEARNING OBJECTIVES

Upon completion of this course, students should be able to:

1. *Design, execute and manage the software development life cycle of a software system using different approaches.*
2. *Create the technical documentation of a project using UML (Unified Modeling Language) and Essence – SEMAT.*
3. *Apply use case patterns in requirements elicitation and analysis.*
4. *Validate the system functionality according to the requirements specified.*

2. PROGRAMME

1. *Software Engineering and the software development problems. Best practices in software development.*
2. *The software process: object-oriented software life cycle approach: Rational Unified Process; SCRUM, DevOps; SecDevOps; Model-Driven Architecture: definition, approach, tools, success stories; process improvement: Capability Maturity Model Integration.*
3. *Requirements specification. Requirements management. Spiral model for requirements engineering.*
4. *Software patterns: introduction of the concept of pattern; study and application of use cases patterns.*
5. *UML - Unified Modeling Language: use cases, modeling the structure, behavior, and architecture. Essence – SEMAT: architecture, dimensions, and key elements.*
6. *Lifecycle of using CASE tools. CASE tools integration.*
7. *Metrics in the software development process.*
8. *Verification and validation of software.*

3. COHERENCE BETWEEN PROGRAMME AND OBJECTIVES

1. Contents 1, 2, 3, 4, 5, 6, 7 and 8 are consistent with Objective 1 because they focus on different approaches.

2. Contents 3, 4, 5 and 6 are consistent with Objective 2, because the many forms of requirement documentation, trained in UML language as well as the tools necessary, are presented.

3. Contents 3 and 4 are consistent with Objective 3, because they teach the patterns for requirements specification in "Patterns for Effective Use Cases".

4. Content 8 is consistent with Objective 4 because it is used the Model V that makes the parallelism between deliveries of the software development process and delivery of the testing process.

4. MAIN BIBLIOGRAPHY

Mandatory:

1. Textos de apoio e diapositivos das aulas fornecidos pelo docente e disponibilizados na plataforma de e-learning.

2. Ivar Jacobson; Harold "Bud" Lawson; Pan-Wei Ng; Paul E. McMahon; Michael Goedicke. The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons; ACM Books, 2019.

3. Nunes, Mauro; O'Neill, Henrique. Fundamental de UML, 5ª Ed., FCA Editora, 2007.

4. Sommerville, I. (2020). *Engineering software products*. London, UK: Pearson*.*

5. Adolph, Steve; Bramble, Paul. Patterns for Effective Use Cases, Addison-Wesley Pearson Education, 2003.

6. Unified Modeling Language™, Resource Page, www.uml.org.

Recommended:

7. Jacobson, Ivar; Booch, Grady; Rumbaugh, James; The Unified Software Development Process; Addison –Wesly; 1999.

8. Rubin, Kenneth S. Essential Scrum: A Practical Guide to the Most Popular Agile Process, 2012, Addison-Wesley Professional.

9. ONeil, H., Nunes, M., Ramos, P. Exercícios de UML, FCA, 2010.

10. Penzenstadler, B., Betz, S., Venters, C.C., Chitchyan, R., Porras, J., Seyff, N., Duboc, L., & Becker, C. (2018). Everything is INTERRELATED: Teaching Software Engineering for Sustainability. IEEE/ACM 40th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), 153-162.

11. Reis, L. Cagica, L., Silveira, C., Russo, N., & Marques A. (2021). Inovação e Sustentabilidade em Tecnologias de Informação e Comunicação. Lisboa: Silabo, ISBN: 978-989-561-146-1.

12. Santos, Vitor (2018). Criatividade em Sistemas de Informação. Lisboa, Editora FCA. ISBN: 978-972-722-891.

13.Sommerville, Ian. Software Engineering (10th edition). Slides disponíveis em: https://software-engineering-book.com/slides/.

14. Torres, J., Julio, D., Silveira, C., & Reis, L. (2022). Diabetes Tracker and Volunteer+ Software Engineering for Sustainability. In L. Reis, L. Carvalho, C. Silveira, & D. Brasil (Ed.), Digitalization as a Driver for Smart Economy in the Post-COVID-19 Era (pp. 198-227). IGI Global.

5. TEACHING METHODOLOGIES (INCLUDING EVALUATION)

Teaching methodologies:

1. *Active learning*
2. *Lecture*
3. *Project*
4. *Case studies*
5. *Problem-solving.*

Evaluation methodologies:

***Continuous evaluation**: The student is required to attend at least 1/2 of the classes to be evaluated during the continuous assessment period. Students with worker-student status are not required to attend.*

80% (16/20) - Activities carried out preferentially during classes, involving individual/group work/development of a project in conjunction with the Curricular Unit (CU) Internet Programming. Students with worker-student status will have to carry out these activities, even outside of class, to be evaluated in this component.

20% (4/20) - Delivery of the pdf report and prototype related to the Project in conjunction with the CU Internet Programming. The prototype has a weight of 2 values. Presentation and defense (mandatory).

Delivery date: January 6, 2024.

*Presentation date (**individual**): January 9 or 12, 2024.*

The evaluation in each of the Curricular Units is done independently, based on the previously defined elements and on the information obtained in class, where guidance, control, supervision and evaluation of the developed project and its progress is done. Students who have passed/are not attending the UC Programming for the Internet, will have to develop and deliver a prototype of the application (development platform of the student's choice and approved by the teacher).

***Exam (appeal season)**: for the student who has not been successful in the continuous evaluation or has not taken it.*

100% (20) - Written test.

6. COHERENCE BETWEEN TEACHING METHODOLOGIES AND OBJECTIVES

1. *Active learning is consistent with the objectives because it allows students to develop collaborative skills, increase interest in the course contents, and introduce creativity and critical thinking that can be applied both in the requirements specification and implementation phases of software projects.*
2. *Lectures are consistent with the objectives due to the need to provide students with the theoretical contents, including the various aspects related to the process models, patterns, metrics, CMMI process improvement.*
3. *Project development is consistent with the objectives since it covers all stages of the software development process, requiring the practical application of all concepts covered throughout the semester to a practical case.*
4. *Case studies are consistent with the objectives since they allow the analysis of other projects documentation, analysis of case studies of companies that use certain approaches (eg MDA), see CMMI certified companies, among others.*
5. *Problem solving is consistent with the objectives since the application of theoretical concepts to solve true to life practical exercises related to the study (system modelling, applying patterns, design tests), helps consolidate the concepts, highlighting the student's expertise.*

7. ATTENDANCE

The student is required to attend at least 1/2 of the classes to be evaluated in the continuous evaluation and in the normal season exam. Students with worker-student status are not required to attend.

8. CONTACTS AND OFFICE HOURS

***Contacts**: Office 21 | e-mail: mclara@ipg.pt*

***Office hours**: Tuesday from 9am to 11am; 2:30 pm to 5 pm; or schedule by email.*

DATE: September 19, 2023 Clique