Politécnico de Viseu
Tecnologia e Gestão Viseu

FICHA ECTS
Engenharia de Software I
2021-22

Caraterização da Unidade Curricular / Characterisation of the Curricular Unit

Designação da Unidade Curricular / Curricular Unit:	[31821481] Engenharia de Software I [31821481] Software Engineering I
Plano / Plan:	Licenciatura em Engenharia Informática V2
Curso / Course:	Licenciatura em Engenharia Informática Computer Sciences Engineering
Grau / Diploma:	Licenciado
Departamento / Department:	Departamento de Informática
Unidade Orgânica / Organic Unit:	Escola Superior de Tecnologia e Gestão de Viseu
Área Científica / Scientific Area:	Ciências Informáticas
Ano Curricular / Curricular Year:	2
Período / Term:	S2
ECTS:	4.5
Horas de Trabalho / Work Hours:	0119:30

Horas de Contacto/Contact Hours:

(T) Teóricas/Theoretical:	0019:30	(TC) Trabalho de Campo/Fieldwork:	0000:00
(TP) Teórico-Práticas/Theoretical-Practical:	0000:00	(OT) Orientação Tutorial/Tutorial Orientation:	0000:00
(P) Práticas/Practical:	0026:00	(E) Estágio/Internship:	0000:00
(PL) Práticas Laboratoriais/Pratical Labs:	0000:00	(O) Outras/Others:	0000:00
(S) Seminário/Seminar:	0000:00		

Docente Responsável / Responsible Teaching

[3047] Carlos Augusto Da Silva Cunha

Docentes que lecionam / Teaching staff

[3047] CARLOS AUGUSTO DA SILVA CUNHA

[3105] JORGE ALEXANDRE DE ALBUQUERQUE LOUREIRO

[3374] JOAO CARLOS AZEVEDO SOUSA

Objetivos de Aprendizagem

O1. Efetuar a análise e especificação de requisitos de software.

O2. Preparar a documentação técnica relativa às diferentes fases do projeto de software.

O3. Conhecer e saber aplicar metodologias de gestão de projetos de software.

O4. Implementar as melhores práticas no desenvolvimento de software, através de técnicas, arquiteturas e conceitos que promovem a sua manutenção corretiva, adaptativa e evolutiva.

Learning Outcomes of the Curricular Unit

O1. Perform analysis and specification of software requirements.
O2. Prepare the technical documentation relative to the different phases of the software project.
O3. Identify and apply management methodologies for software projects.
O4. Implement the best practices in software development, using techniques, architectures, and concepts that promote its evolutive maintenance by designing software for accepting new features.

Conteudos Programáticos (Lim:1000)

1 Introdução
1.1 Estado da Arte
1.2 Conceitos básicos

2 Ciclo de vida do Software
2.1 Engenharia de Requisitos
2.2 Análise
2.3 Desenho
2.4 Arquitetura
2.5 Desenho de Componentes
2.6 Interfaces

3 Metodologias de Desenvolvimento de Software
3.1 Metodologias Clássicas
3.2 Metodologias Ágeis

4 UML
4.1 Diagramas de Casos de Uso
4.2 Diagramas de Classes
4.3 Diagramas de Objetos
4.4 Diagramas de Interação
4.5 Diagramas de Atividade

5 Arquiteturas de Software
5.1 Modularidade
5.2 Reutilização
5.3 APIs e Frameworks
5.4 Facetas Transversais

Syllabus (Lim:1000)

1 Introduction
1.1 State of the Art
1.2 Basic Concepts

2 Software Lifecycle
2.1 Requirements Engineering
2.2 Analysis
2.3 Design
2.4 Architecture
2.5 Design of Components
2.6 Interfaces

3 Software Development Methodologies
3.1 Traditional Methodologies
3.2 Agile Methodologies

4 UML
5.1 Use Cases Diagrams
5.2 Class Diagrams
5.3 Object Diagrams
5.4 Interaction Diagrams
5.5 Activity Diagrams

5 Software Architecture
6.1 Modularization
6.2 Reuse
6.3 APIs e Frameworks
6.4 Crosscutting Concerns

Metodologias de Ensino (Avaliação incluída; Lim:1000)

Aulas de caráter essencialmente prático, com apelo constante à participação, desenvolvimento do espírito crítico e de iniciativa, e procura da excelência nos trabalhos realizados.

Ponderação, em todas as épocas de avaliação:
- **trabalhos práticos** (35%, mínimos: 10/20);
- **prova escrita** (50%, mínimos: 10/20);
- **tarefas resolvidas presencialmente em aula** (15%, mínimos para a Época Normal: 12/20).

Parte da classificação dos trabalhos práticos resulta da sua avaliação contínua (apenas em Época Normal). Todos os trabalhos são passíveis de defesa individual.

Teaching Methodologies (Including evaluation; Lim:1000)

The theoretical classes are expositive. The materials cover all learned content and present illustrative examples.
Students apply the theoretical concepts learned in theoretical classes to practical scenarios in the practical classes. In these classes, we use a set of worksheets, which describe the tasks that students must perform and contain a description of the theoretical concepts needed to solve the presented situations.
Evaluation is based on written tests and practical works. The final mark is calculated as follows:
- Written Test: 50% (minimum of 8 points)
- Practical Works: 35% (minimum of 10 points)
- Assessment worksheets: 15% (Normal Epoch: minimum of 12 points in the assessment worksheets)

Bibliografia de Consulta (Lim:1000)

- Roger S. Pressman, Software engineering : A practitioner's approach, Boston, McGraw-Hill, 2005, (004.41 PRE)
- Martin Fowler and Kendall Scott, UML Distilled: Applying the Standard Object Modelling Language, Addison-Wesley, 1997, (004.43 OBJECTOS FOW)
- G. Booch, J. Rumbaugh, and I. Jacobson, The Unified Modelling Language User Guide, AddisonWesley, 1999, (004.41 BOO)
- Mauro Nunes e Henrique ONeill, Fundamental de UML, FCA Editora de Informática, 2001, (004.41 NUN)
- Kevin C. Desouza, Agile information systems : Conceptualization, construction, and management, Elsevier, 2007, (004.952 AGI)
- Rui Feio, Gestão de projectos com o microsoft project 2007, FCA, 2008, (004.42 FEI)
- Thinking in Patterns with Java, http://www.bruceeckel.com

- Ian Sommerville, Software Engineering (10th. ed.), Pearson, 2015.

- I. Sommerville, Engineering software products: An introduction to modern software engineering, Pearson Education, 2020.

Bibliography (Lim:1000)

- Roger S. Pressman, Software engineering : A practitioner's approach, Boston, McGraw-Hill, 2005, (004.41 PRE)
- Martin Fowler and Kendall Scott, UML Distilled: Applying the Standard Object Modelling Language, Addison-Wesley, 1997, (004.43 OBJECTOS FOW)
- G. Booch, J. Rumbaugh, and I. Jacobson, The Unified Modelling Language User Guide, AddisonWesley, 1999, (004.41 BOO)
- Mauro Nunes e Henrique ONeill, Fundamental de UML, FCA Editora de Informática, 2001, (004.41 NUN)
- Kevin C. Desouza, Agile information systems : Conceptualization, construction, and management, Elsevier, 2007, (004.952 AGI)
- Rui Feio, Gestão de projectos com o microsoft project 2007, FCA, 2008, (004.42 FEI)
- Thinking in Patterns with Java, http://www.bruceeckel.com

- Ian Sommerville, Software Engineering (10th. ed.), Pearson, 2015.

- I. Sommerville, Engineering software products: An introduction to modern software engineering, Pearson Education, 2020.

Observações

«Observações»

Observations

«Observations»

Observações complementares