deti universidade de aveiro
departamento de eletrónica, telecomunicações e informática

TQS Dossier Pedagógico

I. Oliveira, em 2022-03-07.

Docentes: Ilídio C Oliveira (ico@ua.pt), Joaquim Sousa Pinto

Créditos: 6 ECTS (~1 dia de dedicação/semana)

Resumo: Apresenta o plano de estudos da unidade curricular 45426- Teste e Qualidade de Software (TQS), 2021/22.

1 Conteúdo da disciplina

1.1 Objetivos de aprendizagem

O objetivo geral da unidade curricular (UC) é fornecer aos alunos um conjunto de princípios e práticas que lhes permita definir e aplicar um processo de garantia de qualidade de software que seja sistemático, orientado aos testes e, tanto quanto possível, automático.

No final da disciplina, os alunos deverão ser capazes de:

- reconhecer a gestão de qualidade de software como um processo, apresentando os papéis e atividades envolvidas.
- descrever diferentes níveis e tipos de testes, a forma de os operacionalizar e o respetivo contributo no processo de garantia de qualidade.
- conceber e escrever casos de teste a partir de especificações de requisitos.
- recorrer a ferramentas para a verificação de software, incluindo utilização de testes unitários.
- aplicar as práticas de desenvolvimento orientado por testes (TDD) em pequenos projetos.
- aplicar técnicas de análise estática de software e participar em sessões de revisão de código.
- montar e usar um ambiente de integração contínua para desenvolvimento em equipa, baseado em ferramentas automáticas, com integração de testes.
- propor, montar e usar uma solução de gestão de defeitos e documentação de testes.

1.2 Conteúdos programáticos

O programa indicativo da unidade curricular para esta edição é o seguinte:

- Conceitos em qualidade de software
    - Fatores de qualidade
    - Elementos de um sistema de garantia de qualidade
- Revisão e melhoria do código
    - Métricas para análise de código
    - Análise estrutural apoiada em ferramentas
    - Inspeção e revisão de código
    - Reconstrução (*refactoring*) para melhorar o código existente
- Métodos de teste de software
    - Níveis de teste de verificação (unidade, integração, sistema)
    - *Frameworks* para testes unitários
    - Estratégias para o teste de código de caixa-aberta
    - Testes de carga e desempenho
    - *Frameworks* para automatizar teste de aplicações web
- O processo de teste
    - Integração de testes em metodologias ágeis
    - Planeamento, execução e gestão de testes
    - Desenvolvimento orientado por testes (TDD)
- Boas práticas de equipa
    - Seguimento e gestão de defeitos

    - Gestão de configurações
    - Manuais de estilo e de práticas partilhadas
- *Pipelines* de entrega contínua
    - Integração contínua: a linha de montagem
    - Provisão automática de ambientes virtuais para teste e integração

1.3 Modelo de Ensino/Aprendizagem

As aulas estão organizadas em atividades semanais, incluindo o enquadramento dos assuntos na aula TP e a realização de exercícios de consolidação nas aulas P.

As aulas P serão usadas para atividades laboratoriais guiadas e, na parte final do semestre, para projetos de grupo.

1.4 Pré-requisitos

Domínio da linguagem de programação Java.

A utilização do portátil pessoal é recomendada.

1.5 Bibliografia recomendada

Não há um livro de texto obrigatório.

Referências principais:

- J. Humble and D. Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation: Pearson Education, 2010. → OReilly
- P. Farrell-Vinay, Manage Software Testing: Taylor & Francis, 2008.

Referências complementares recomendadas:

- B. Hambling, Ed., Software testing : An ISTQB-ISEB Foundation Guide (2nd). British Informatics Society Limited and The Chartered Institute for IT, 2010. [Guia usado para a certificação ISTQB-ISEB]
- J. Bloch, Effective Java, 3rd ed.: Addison-Wesley Professional, 2017. → OReilly
- S. McConnell, Code complete, 2nd ed. Microsoft Press, 2004. → OReilly
- R. Martin, Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall; 2008. → OReilly
- R. Pressman and B. Maxim, Software Engineering: A Practitioner's Approach, 8th ed. McGraw-Hill Science/Engineering/Math, 2014.

2 Regras gerais de funcionamento

Aplicam-se as regras gerais de funcionamento das aulas descritas no Regulamento de Estudos de Licenciaturas e Mestrados da Universidade de Aveiro.

Nos termos regulamentares, a propriedade intelectual dos trabalhos realizados no contexto das aulas é da Universidade de Aveiro.

2.1 Avisos e disponibilização de materiais

Os avisos da unidade curricular serão “afixados” no sistema de eLearning em uso na UA.

A disponibilização dos materiais pedagógicos será feita no sistema de eLearning e em recursos devidamente hiperligados a partir da página da unidade curricular.

2.2 Regras de avaliação

A disciplina funciona (por omissão) em regime de **Avaliação Discreta**, com três momentos de avaliação:

1. (Pr1) Trabalho individual de acompanhamento [20%]
   - 1 trabalho de casa individual
   - portfolio individual com as entregas dos guiões das aulas P
2. (Pr2) Projeto grupo [40%]
3. (TP) Exame escrito (individual) [40%]

A fórmula de cálculo para determinar a nota final é a seguinte:

Nota = 0,40 TP + 0,20 Pr1 + 0,40 Pr2

A avaliação em recurso pode incidir sobre a componente prática (através de uma nova entrega a combinar com o docente) e/ou sobre a componente teórico-prática (exame escrito). O recurso da prática substitui todas as componentes práticas do período “Normal”.

Nota mínima, por componente (TP ou P): 7 (sete) valores.

Nos termos regulamentares, os alunos podem, em alternativa, optar no início do ano letivo pelo regime de Avaliação Final (eventualmente mais adequado aos Trabalhadores-estudantes que não frequentam as práticas). Nesse caso, as componentes de avaliação são estas:

componentes da UC	P	TP
peso da componente	**50%**	**50%**
Prova escrita / Written test	0%	50%
Desenvolvimento e defesa de projeto / Project assignment	50%	0%

2.3 Regime de faltas

Marcação de faltas não aplicável à TP. Presença obrigatória nas P (conforme estipulado no Regulamento de Estudos da UA).

2.4 Política de colaboração

Os alunos são incentivados a colaborar, entreajudando-se no seu percurso académico. Essa colaboração, não pode, no entanto, configurar situações de plágio ou de desonestidade académica. Pressupõe-se que todo o trabalho entregue foi efetivamente realizado pelos autores nele identificados. Caso exista evidência em sentido contrário, o trabalho é anulado e o incidente participado aos órgãos competentes para eventual procedimento disciplinar.

Sempre que um trabalho ou outro elemento de avaliação tem natureza individual, deve ser feito pelo próprio aluno, sem copiar conteúdo de colegas ou de outras fontes, designadamente da Internet. A utilização de bibliotecas (de código) externas deve ser confirmada com o docente.

Dentro de cada grupo, o trabalho deve, naturalmente, ser sempre partilhado e contribuído por todos. Os alunos devem participar ao docente as situações em que algum membro do grupo não faz as tarefas que lhe foram de comum acordo atribuídas.

Cumulativamente, os alunos comprometem-se a observar a Carta de Conduta dos Estudantes da Universidade de Aveiro.

2.5 Entregas

Os trabalhos são lançados no sistema de e-Learning da UA e é aí que as entregas devem ser feitas (em formato eletrónico).

A entrega de código deve ser feita através da utilização de um sistema de repositório de código a designar.

Nos projetos de programação, é requerida a adesão às práticas de programação definidas para a disciplina.