 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame de Época Normal	Ano letivo 2022-2023	Data 07-02-2023
	Curso Licenciatura em Engenharia Informática	Hora 10:00	
	Unidade Curricular Engenharia de Software II	Duração 90 minutos	

NOTAS IMPORTANTES


- O presente exame é sem consulta!
- Antes de iniciar a leitura do enunciado, preencha as folhas de exercício que lhe foram fornecidas com a sua informação pessoal.
- Coloque o seu número e nome no enunciado que deverá entregar no final do exame. Contudo, todas as respostas devem ser dadas na folha de exame e não no enunciado.
- Em cada questão poderá encontrar a respetiva cotação, em valores, entre 0 – 20.
- O teste é composto por um grupo de questões de escolha múltipla e, eventualmente, verdadeiro/falso. Para cada questão, assinale a(s) resposta(s) que lhe parecer(em) mais correta(s). Deverá ainda, justificar todas as respostas cuja opção que selecionou corresponde á seguinte afirmação: “Nenhuma das opções (...)”. A não justificação implica a não avaliação da questão.
- Deverá justificar as afirmações que considerar falsas. A não justificação implica a não avaliação da questão.
- Relativamente às questões de escolha múltipla será descontado (1/4)*cotação da pergunta por cada resposta errada. Nas questões do tipo verdadeiro/falso o desconto passará para 1/2.
- Nas questões em que haja mais do que uma opção para escolha, a cotação será dada de acordo com as seguintes regras: i) a cotação da questão será dividida equitativamente pelo número de opções corretas; ii) por cada opção errada selecionada, será descontado (1/4)*cotação da alínea;
- APENAS SE ESCLARECEM DÚVIDAS RELACIONADAS COM DIFICULDADES DE INTERPRETAÇÃO DO ENUNCIADO OU CASO SE VERIFIQUE GRALHAS NA ELABORAÇÃO DO MESMO.
- Nota mínima: 9,5 valores
- Nota mínima para avaliação contínua. 7,5 valores

Os docentes: Cristóvão Sousa (cds@estg.ipp.pt)
Ricardo Ferreira (rmf@estg.ipp.pt)

GRUPO I – Escolha Múltipla


1. (Cotação: 1,5 valores)

Qual das seguintes opções pode ser considerada uma métrica relacionada com o processo de desenvolvimento de software?

- a) Número de "User Stories" realizadas por sprint. 
- b) Número de linhas de código por sprint.
- c) Número de relações do tipo "herança" entre componentes de software.
- d) Cyclomatic Complexity
- e) Nenhuma das opções apresentadas é correta.


2. (Cotação: 1,5 valores)

A adoção de um processo de Gestão de Configuração de Software pretende:

- a) Controlar a evolução e integridade de um produto de software e da identificação dos seus elementos;
- b) Controlar e gerir a mudança, fornecer mecanismos de verificação e reporte da configuração do produto de software;
- c) Identificar, controlar, acompanhar e auditar os componentes de software à medida que o produto de software evolui;
- d) Todas as opções apresentadas estão corretas 
- e) Nenhuma das opções apresentadas está correta


3. (Cotação: 1,5 valores)


Prototipagem não é, para muitos, considerada um modelo de desenvolvimento completo, mas uma abordagem para mitigar o risco no contexto de um modelo7metodologia mais completo. Tipicamente os modelos que não usam prototipagem são:

- a) Modelos Ágeis
- b) Modelo Espiral;
- c) Modelos Incrementais
- d) Nenhuma das opções apresentadas é correta. 

4. (Cotação: 1,5 valores)

No âmbito da implementação de um plano SCM (*Software Configuration management*) qual destes papeis inerentes à metodologia SCRUM, poderia assumir a responsabilidade de CCB (*Configuration Control Board*) Leader?

- a) SCRUM Master
- b) Product Owner 

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame de Época Normal	Ano letivo 2022-2023	Data 07-02-2023
	Curso Licenciatura em Engenharia Informática	Hora 10:00	
	Unidade Curricular Engenharia de Software II	Duração 90 minutos	

c) Dev Team e Product Owner

5. (Cotação: 1,5 valores)

De acordo com a metodologia SCRUM, selecione a(s) opções/opção que correspondem a atividades que não ocorrem na fase de planeamento?

- a) Definição da arquitetura do sistema;
- b) Seleção/definição das convenções associadas o desenvolvimento do produto de software (exemplo: *coding style*)
- c) Definir o modelo de desenvolvimento;
- d) Selecionar a tecnologia
- e) Nenhuma das opções apresentadas é correta.

6. (Cotação: 1,5 valores)

Os modelos evolutivos procuram colmatar o facto de, frequentemente, o cliente não saber em concreto o que quer. Contudo, estes modelos encerram outros riscos, nomeadamente:

- a) Gerir o alargamento do âmbito do sistema
- b) Controlar a documentação
- c) Conceber, antecipadamente, um plano de testes
- d) Gerir as *baselines* do projeto
- e) Nenhuma das opções apresentadas

7. (Cotação: 1 valores)

Considere um projeto em que: i) o orçamento não é folgado; ii) o tempo para entrega do produto final é flexível; iii) Os requisitos estão vagos, contudo o risco associado ao sucesso do projeto continua a ser uma preocupação. Que modelo de desenvolvimento de software considera mais adequado?

- a) Ágil
- b) Incremental
- c) Espiral
- d) Nenhuma das opções apresentados

GRUPO II – Questões Abertas

1. (Cotação: 3,5 valores)

O esquema da figura x, representa os *stages* de um *pipeline* implementado usando a ferramenta *GitLab* de modo a assegurar atividades de Integração Contínua.




Figura 1- Esquema de pipeline CI/CD implementado no GitLab

Para cada um desses *stages*:

- a) Caracterize cada uma dos *stages* e indique os comandos *gradle* subjacentes a cada *stage*;

Inspection: verificação estática de código. Verificação de boas práticas de desenvolvimento de código; verificação de convenções e coding Styles verificação de princípios de OO design; documentação; ...
Gradle: `gradle check`

BlackBox Testing: verificação dinâmica do comportamento do(s) componente(s) funcionais através de testes de software

	ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	Tipo de Prova	Ano letivo	Data
		Exame de época de NORMAL	2019/2020	29-01-2020
		Curso	Hora	
		Licenciatura em Engenharia Informática	10:00	
		Unidade Curricular	Duração	
		Engenharia de Software II	90 minutos	

Gradle: gradle test

***WhiteBox Testing:** A única diferente entre BB Testing e WB Testing, a este nível, poderá ter a ver com os testes que iremos executar. De resto tudo de mantém idêntico ao stage anterior*

***Report:** Corresponde à geração dos artefactos resultantes dos do stage de verificação e do stage de testing. Depende da configuração dos Plug-ins no gradle e da respetiva configuração das gradle tasks*

***Build:** Verifica, testa, compila e agrega todos os componentes gerando o executável.*

Gradle: gradle build


- b) Quais dos *stages* representados incluem práticas verificação e quais os que incluem práticas de validação do produto de software;

- Não há práticas de validação, apenas de verificação.

- c) Concretamente, qual dos *stages* corresponde a uma verificação funcional;

- Os stages de testing e build (volta a fazer testes)

2. (Cotação: 3 valores)

No âmbito garantia da qualidade de software, identifique e explique 3 práticas para assegurar componentes de software acoplamento fraco com coesão forte. 

- Uso de interfaces como forma privilegiada de comunicação entre os componentes;
- Evitar métodos demasiados extensos e demasiados métodos (não vamos considerar aqui o método de acesso);
- Evitar que os componentes tenham a necessidade de aceder a dados de outros componentes continuamente;
- Evitar dependências de estado (exemplo: composição e agregação).

3. (Cotação: 3,5 valores)

Explique de que forma a prática SCRUM de estimar as *User Stories* contribui para a garantia da qualidade do processo de desenvolvimento de software.

1. O CMMI é um modelo standard para a gestão da qualidade cujo foco está no melhoramento contínuo dos processos de desenvolvimento de SW. Por sua vez, estimar corresponde a uma área de prática no âmbito do CMMI.

2. Entre várias coisas, estimar no âmbito do CMMI prevê o desenvolvimento de estimativas no sentido de monitorizar o esforço e duração associada aos artefactos que compõem a solução. Adicionalmente o CMMI indica que se deve medir este esforço e acompanhar a estimativa de trabalho ao longo do projeto (monitorizar).

*3. Estimar as user stories, permite avaliar o esforço associado à implementação de determinadas features no projeto, permite ainda monitorizar o desenvolvimento das mesmas através do **BurnDown Chart**. Deste modo poderemos afirmar que estimar user stories no âmbito da metodologia SCRUM é uma prática alinhada com os princípios CMMI de garantia de qualidade.*

Outra forma de construção da resposta:

- Falar da ligação ao CMMI como modelo standard para gestão da melhoria continua do processo de DEV. Em SCRUM estimamos então estamos alinhados com os princípios de garantia de qualidade do CMMI
- Falar que é uma forma de gestão colaborativa de risco → Toda a gente está consciente do risco!
- Falar sobre controlo e monitorização via BurnDown Chart que permite tomar decisões antecipadamente caso alguma coisa esteja a correr mal → passamos para um nível de proatividade
- Melhor planeamento envolvendo os stakeholders