Teste 2

Métodos de Desenvolvimento de Software
2012/13
Segundo Teste – 13 de Dezembro de 2012
Departamento de Informática
Universidade Nova de Lisboa
(duração 1h30)

– Sem consulta –

Leia com atenção a informação constante desta página, enquanto espera a indicação do docente para começar a resolução do teste.

Este enunciado é composto por:
Uma página de Rosto (esta)
11 páginas de enunciado.

O exame é composto por sete grupos. Existem três tipos de perguntas:

I. Resposta múltipla para seleção de apenas uma alínea – Uma resposta errada desconta metade do valor dessa pergunta na cotação total do teste (VALOR DA PERGUNTA) / 2
II. Múltiplas afirmações para indicar todas as que se aplicam – cada escolha errada desconta um valor da cotação total do teste calculado da seguinte forma:
2 X (VALOR DA PERGUNTA) / (NÚMERO DE RESPOSTAS)
III. Repostas de caixa aberta.

Todas as perguntas devem ser respondidas o próprio enunciado assinalando com um X a opção/opções pretendidas. (caso haja engano fazer uma bola por cima do X errado e voltar a colocar o X na opção pretendida, em caso de dúvida perguntar ao docente).

Todas as páginas deverão ter o nome e número de aluno para ser consideradas para avaliação.

No fim de 1h30 de exame o docente **recolherá o enunciado e a folha de respostas**.

Boa Sorte!

Parte I – Modelos e Metamodelos

1. O que é um modelo?
(escolher todas as afirmações verdadeiras)

a) Uma realidade, objecto específico ou instância do sistema
b) Uma representação de um sistema que tem em atenção todos os detalhes deste sistema
c) Uma qualidade ou característica que implica reuso
d) Uma abstração e simplificação da realidade
e) Uma forma de entender e analisar um sistema que pode ser usada para automatizar o processo de desenvolvimento de software
f) É informação a um determinado nível de abstração
g) É uma forma de lidar com a complexidade
h) Um desenho

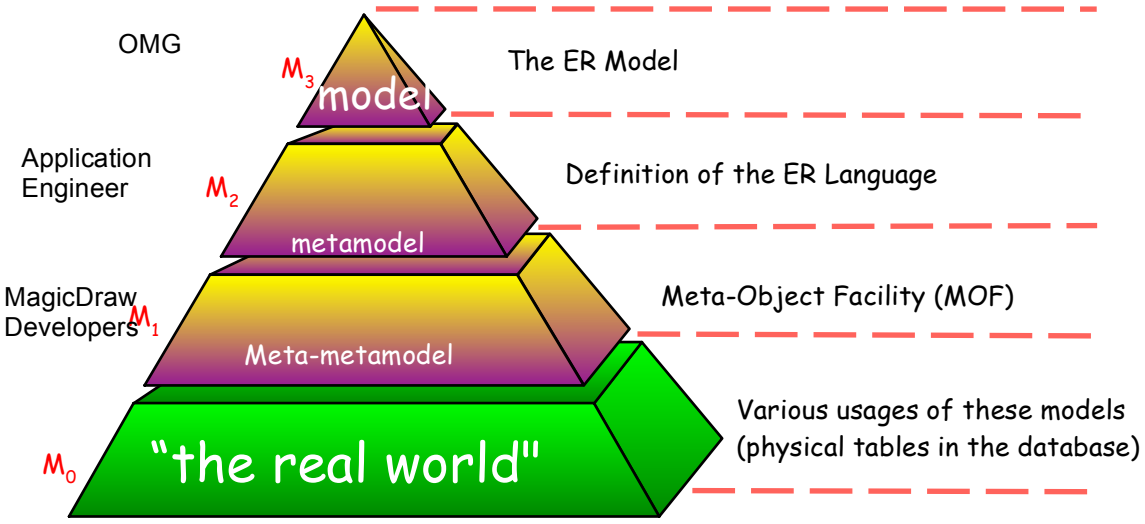
2. Um metamodelo é:
(escolha todas as afirmações que se aplicam)

a) Um modelo que descreve um modelo
b) Uma instância de um programa
c) Uma explicação sobre um conjunto de modelos todos representados de uma maneira diferente
d) Uma definição precisa das regras de construção necessárias para construtores e regras necessárias para criar modelos com uma determinada representação

3. Quais são as vantagens de se estruturar e estudar as várias camadas de metamodelação?
(escolha apenas uma, a melhor resposta verdadeira)

a) É um conceito abstracto que especifica o tipo de relação entre elementos
b) Relação é uma seta entre dois elementos
c) É um concito abstracto que especifica algum tipo de relação entre dois elementos
d) Define uma associação entre elementos

4. Arquitectura de metamodelos da organização OMG.

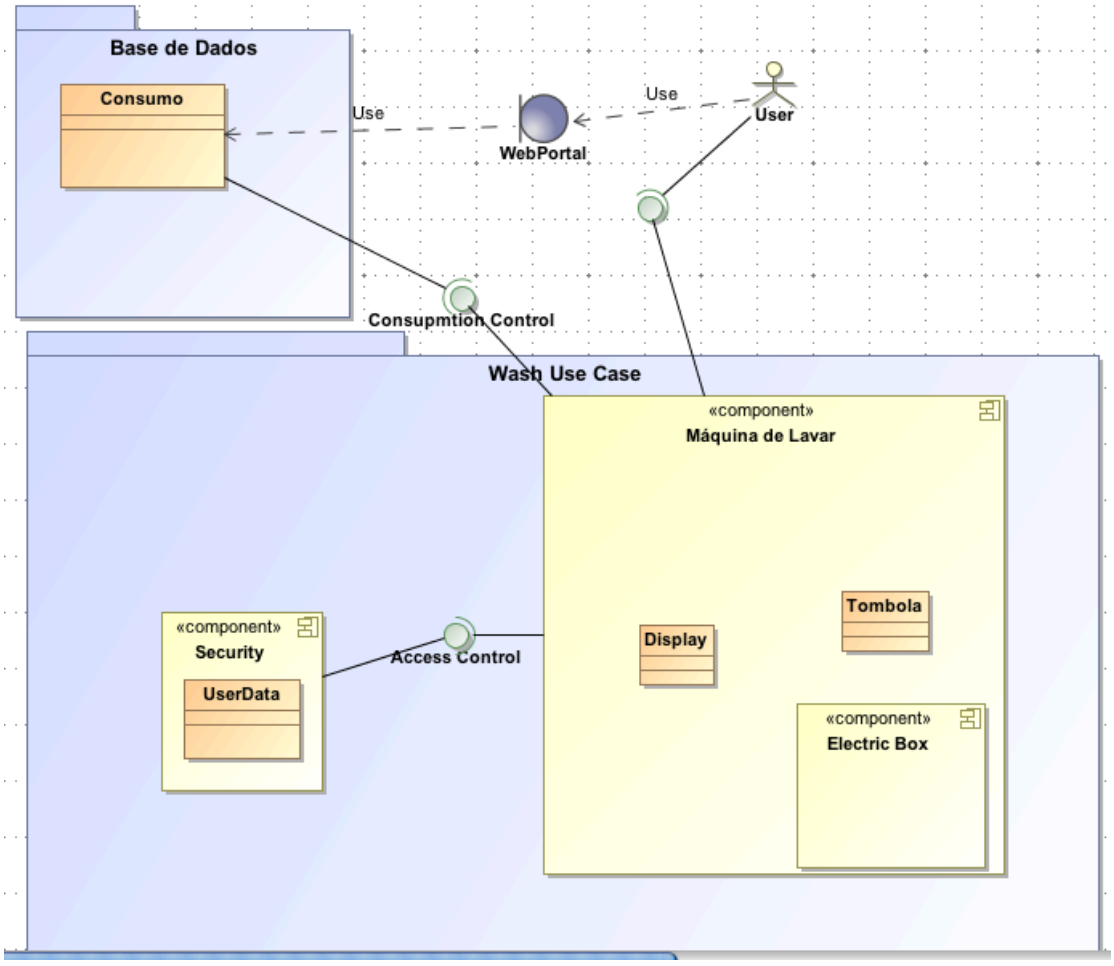


Explique o diagrama e Identifique, caso existam, as incorreções (justificando-as igualmente).

Parte II - Diagramas de Componentes

1. Indique **todas as afirmações falsas** sobre diagramas de componentes:

a) Um componente é uma parte modular do sistema que encapsula o seu conteúdo
b) Um componente pode ser substituível por outro desde que suporte o mesmo protocolo
c) Um componente pode ser algo que é instanciado em runtime como uma Enterprise JavaBean
d) Um componente pode assumir diferentes estereótipos: Application, datastore, document, entity, executable, file, infrastructure, libray, process, source code, specification etc
e) Acoplamento baixo significa desenhar componentes de modo a que o fluxo de informação ocorra dentro de um componente
f) Coesão alta significa que o componente do domínio é normalmente composto por um conjunto de classes que colaboram entre si
g) Classe de uma mesma hierarquia (herança ou composição) não são representáveis todos no mesmo componente
h) As interfaces oferecem serviços que podem ser manipulados por atores
i) Na notação "Lolipop", um componente oferece um serviço tendo para isso um símbolo semelhante a uma mão aberta, e usa um serviço com a notação correspondente à mão fechada



2. Identifique todas as razões pelas quais o diagrama seguinte está incorreto.

Marque na figura as incorreções identificadas (com um código de referência: A,B,C,...) e Justifique detalhadamente em baixo para cada um deles.

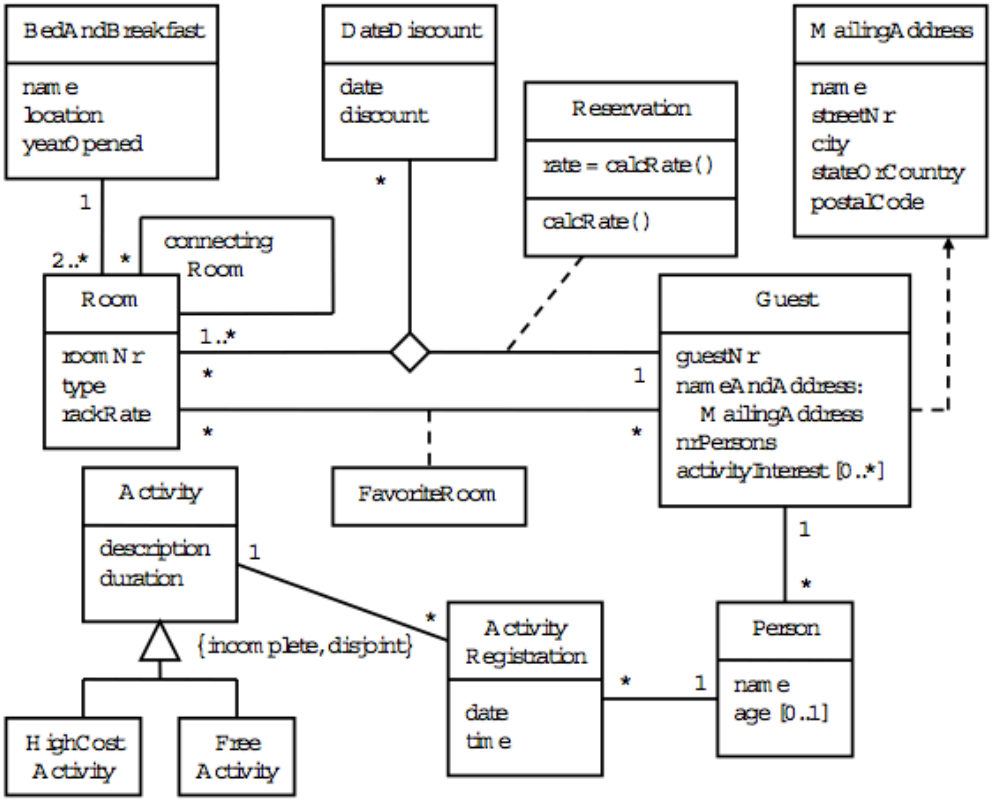
Parte III – Diagramas de Estados

1. Um diagrama de estados:
(escolha as resposta **falsas**)
   a) representa os diferentes estados pelos quais um objecto passa durante o seu ciclo de vida no sistema
   b) representa como um objecto responde a eventos
   c) pode ter estados compostos, não pode representar
   d) possui representação visual com nós e arestas, em que os primeiros representam atividades e os segundos transições desencadeadas por simples conclusão das primeiras
   e) possui quatro tipos de eventos: Sinais, Invocação, passagem de tempo e mudnça de estado
   f) um estado pode incluir ações de entrada, ações de saída, atividades, transições internas e ações que são deferidas
   g) uma transição pode ser descrita por **evento [condição com guarda] / Ação**

2. Desenhe o diagrama de estados para descrever o estado civil de uma pessoa.

Parte IV – De Orientação por objetos para Relacional

1. Considere o seguinte diagrama de classes e regras OCL correspondentes.



context BedAndBreakfast inv:
BedAndBreakfast.allInstances()->forAll(b1, b2 |
b1.name = b2.name and b1.location = b2.location implies b1 = b2)

Repetem-se as regras do mesmo estilo para:
-- name location → BedAndBreakfast
-- roomN r name location → Room
-- guestNr → Guest
-- date → DateDiscount
-- name guestN r → Person
-- description → Activity
-- name guestNr date time → ActivityRegistration

Traduza o esquema anterior para tabelas ER, não se esquecendo de identificar claramente as chaves primárias e estrangeiras.

Parte V – Engenharia de Software

1. Escolha as afirmações correctas:
   a) A Engenharia de Software é uma disciplina preocupada com todos os aspectos da produção de Software desde a especificação de um sistema até à sua manutenção
   b) A Engenharia de Software procura definir Standards de interoperabilidade de produtos voltados para a indústria
   c) Há vários projectos que não usam Engenharia de Software e são bem sucedidos, sendo inclusivamente repetíveis
   d) Problemas no desenvolvimento do Software associados à saida de profissionais da equipa de desenvolvimento, ou falta de dinheiro para continuar o projecto, dizem respeito à Gestão da empresa e não à Engenharia de Software.

2. Quais são as boas características desejáveis num produto de Software?
(Assinale todas as que se aplicam)
   a) Manutenção
   b) Visibilidade
   c) Segurança
   d) Dependability
   e) Ter mercado
   f) Patenteável
   g) Eficiência
   h) Usabilidade

3. Quais são as actividades de processo fundamentais em Engenharia de Software?
(Escolher a mais correcta)
   a) Documentação, Implementação, Testes, Entrega
   b) Especificação, Desenvolvimento, Validação, Evolução
   c) Levantamento de Requisitos, Modelação em UML, Implementação, Entrega
   d) Análise, Implementação, Entrega, Manutenção
   e) Nenhuma das anteriores

4. O código de Ética profissional da Engenharia de Software:
(escolha as afirmações correctas)
   a) É um conjunto de princípios que estabelecem, de um modo geral, um standard para o comportamento esperado de um profissional em engenharia de Software
   b) Inclui como dever representar e honrar o local de formação académica
   c) Tem como responsabilidade garantir: confidencialidade, competência, defesa de direitos de propriedade e que não há uso dos sistemas com intenções criminosas
   d) Promover vigilância das acções do cidadão
   e) Sugere o associativismo com vista a fazer valorizar a profissão com respeito a outras associações profissionais, o que leva à defesa dos direitos dos Engenheiros.

5. No contexto dos modelos de processos de desenvolvimento:
(indique quais das seguintes afirmações **não são** verdadeiras)

   a) As "task Regions" de um modelo em espiral são: comunicação ao cliente, planeamento, análise de risco, engenharia, construção e entrega, avaliação.
   b) O modelo de Rapid Application Development para ser aplicado e mais vantajoso que o modelo em cascata exige mais recursos e mais tempo disponível para o projeto
   c) O modelo em cascata é especialmente adequado a desenvolvimento de produtos de grande dimensão e inovadores
   d) O Modelo de Prototipagem permite entregar produtos operacionais a cada iteração do ciclo, embora use tecnologia que ignora a performance e a qualidade. Esta característica torna-o comparável ao modelo em Espiral

Parte VI – Agile process development

1. Para que tipos de sistemas são os processos Ágeis mais bem sucedidos?
(Assinale as falsas)
   a) Para desenvolvimento de produtos de pequena e média escala
   b) Para desenvolvimento de produtos de grande escala, onde é crucial conjugar esforços de muitas equipas
   c) Onde os clientes estão claramente envolvidos no processo de desenvolvimento
   d) Onde a existência documentação a todo o momento do processo e a capacidade de se realizar auditorias ao software desenvolvido, é exigido pelos organismos que regulam o mercado
   e) Onde não se pode sentar ao lado do cliente e não se quer documentação

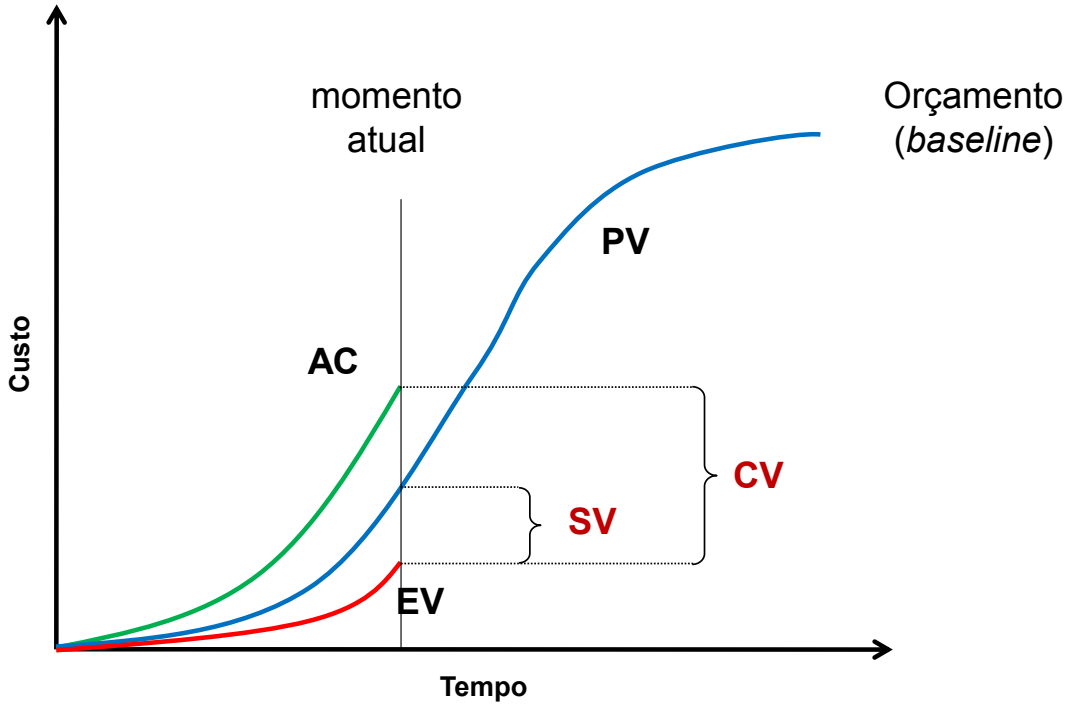
2. Princípios dos métodos Ágeis
(assinale as falsas)
   a) Envolvimento do cliente
   b) Envolvimento dos Stakeholders
   c) Entregas incrementais
   d) Centrar nas pessoas e não no processo
   e) Não documentar
   f) Manter simplicidade
   g) Abraçar a mudança
   h) Capacidade de previsão

3. Que questões devemos fazer para avançar para um método Ágil?
(assinale as verdadeiras)
   a) É a estratégia de entregas incrementais realista?
   b) Que tipo de sistema está a ser desenvolvido?
   c) Qual é o tempo de vida esperado para o sistema?
   d) Como está organizada a equipa de desenvolvimento?
   e) Quão grande é o sistema a ser desenvolvido?

4. Características do extreme programming (XP) ?
(quais as falsas)
   a) Requisitos expressos em cenários
   b) Desenvolvimento centrado no testar-primeiro
   c) Cada programador desenvolve de um modo ágil a sua parte e integra, quando acaba, com o resultado dos outros programadores
   d) Ajuda a treinar inexperientes

5. Indique quais as verdadeiras:
   a) Quando uma característica desejável (feature) é identificada, são logo escritos os testes ao código que implementam essas características, mesmo antes de este ser feito. Os testes são automatizados e são logo executados assim que a funcionalidade é adicionada ao sistema.
   b) No desenvolvimento orientado ao teste-primeiro os programadores podem ir por atalhos ao desenvolver os testes criando uma bateria de testes incompletos. Ao mesmo tempo, alguns testes podem ser difíceis de escrever de um modo incremental, e é difícil de estimar a completude do conjunto de testes.
   c) O sprint Scrum é uma unidade planeada para curta duração (3 a 4 semanas) na qual o trabalho a realizar é determinado, as funcionalidades são selecionadas para desenvolvimento, e o software é entregue aos stakeholders.
   d) A limitação à introdução de métodos ágeis em grande empresas
   a) É a de que os gestores podem ter relutância em aceitar o risco de uma nova abordagem. Os procedimentos nas grandes empresas pode ser incompatível com a abordagem informal à documentação praticada nos métodos ágeis. As equipas podem não ter os skills necessários à utilização de métodos ágeis. Resistência cultural se existir uma longa história de desenvolvimento orientado ao planeamento.

Parte VII – Earned Value Management



1. Gestão de Projetos

(Escolha as afirmações verdadeiras)

a) Gerir projetos é garantir que as "red-team" (ou equipas de emergência) estão sempre prontas a atuar
b) O EVM, comparado com a contabilidade tradicional, não nos dá uma imagem do projeto, ou forma de prever resultados na conclusão
c) Ao fazer uma análise de performance de um desvio do prazo, com o valor do Schedule Variance (Earned Value – Planed Value) com valores menores que 0, podemos concluir que o projecto está adiantado relativamente ao prazo
d) Se ao fazer uma análise de performance de um desvio do prazo usando o valor do Cost Variance (Earned Value-Actual Cost) com valores maiores que 0, podemos concluir que o projeto está sobre-orçamentado.
e) O Cost Performance Index (CPI) é usado como uma ferramenta previsão que dá uma medida de produtividade
f) Um Schedule Performance Index (SPI) =1 significa que o prazo do projeto está de acordo com o plano

FIM!