UA/DETI • Modelação e Análise de Sistema
Ilídio C Oliveira (ico@ua.pt)

Last revision: 2019-06-08.

Tópicos de Estudo

Utilize este guia de estudo para revisão para o exame e autodiagnóstico. Este guia não constitui uma lista de questões possíveis; deve ser visto como uma súmula dos tópicos que podem ser abordados.

O que é que o SDLC inclui?

O trabalho do Analista na equipa de desenvolvimento

Principais referências: TP1

- Explique o que é o ciclo de vida de desenvolvimento de sistemas (SDLC)
- Descreva as principais atividades (etapas-chave) dentro de cada uma das quatro fases do SDLC
- Descreva o papel e as responsabilidades do Analista no SDLC

O Unified Process/OpenUP

Referências principais: TP1, TP4

- Descrever a estrutura das OpenUP (fases e iterações)
- Descrever os objectivos do ciclo de vida de cada fase
- Identificar as principais atividades de modelação/desenvolvimento associados a cada fase
- O OpenUP pode ser considerado "método ágil"?
- Porque é que o Unified Process é "orientado por casos de utilização, focado na arquitetura, iterativo e incremental"?

Principais características dos métodos Ágeis

Principais referências: TP11

- Identificar características distintivas dos processos sequenciais, como a abordagem *waterfall*.
- Identificar as práticas distintivas dos métodos ágeis (o que há de novo no modelo de processo, comparando com a abordagem "tradicional"?).
- Discuta o argumento que "A abordagem em cascata tende a mascarar os riscos reais de um projeto até que seja tarde demais para fazer algo significativo sobre eles."
- Identifique vantagens de estruturar um projeto em iterações, produzindo incrementos com frequência
- Caracterizar os princípios da gestão do *backlog* em projetos ágeis.
- Dado um "princípio" (do *Agile Manifest*), explique-o por palavras próprias , concentrando-se na sua novidade (com relação às abordagens "clássicas") e impacto / benefício.

O papel da modelação (visual)

Principais referências: TP01, TP02, TP05

- Justifique o uso de modelos na engenharia de sistemas
- Descreva a diferença entre modelos funcionais, modelos estáticos e modelos de comportamento.
- Enumerar as vantagens dos modelos visuais
- Explicar a organização da UML (classificação dos diagramas)
- Identificar os principais diagramas na UML e seu respetivo "ponto de vista" (perspetiva) de modelação
- Ler e criar Diagramas de Atividades, Diagramas de Casos de Utilização, Diagramas de Classes, Diagramas de Sequência, Diagramas de Estado, Diagramas de Implementação, Diagramas de Pacotes e Diagramas de Componentes.

Modelos de análise

Práticas de engenharia de requisitos

Principais referências: TP02, TP03.

- Distinguir entre requisitos funcionais e não funcionais
- Distinguir entre abordagens centradas em cenários (utilização) e abordagens centradas no produto para a elicitação de requisitos
- Identificar, numa lista, requisitos funcionais e atributos de qualidade.
- Justifique que "a elicitação de requisitos é mais que a recolha de requisitos".
- Identifique requisitos bem e mal formulados (aplicando os critérios S.M.A.R.T.)
- Enumerar os principais recursos das ferramentas / ambientes do software Requirements Management.

A modelação do contexto do problema: modelo do domínio/negócio

Principais referências: TP01, TP05, TP06.

Caraterizar os conceitos do domínio de aplicação

- Desenhe um diagrama de classes simples para capturar os conceitos de um domínio de problema.
- Apresente duas estratégias para descobrir sistematicamente os conceitos candidatos para incluir no modelo de domínio.
- Identificar construções específicas (associadas à implementação) que podem poluir o modelo de domínio (na etapa de análise).

Caraterizar os processos do negócio/organizacionais

- Leia e desenhe diagramas de atividades para descrever os fluxos de trabalho da organização / negócios.
- Identifique o uso adequado de ações, fluxo de controle, fluxo de objetos, eventos e partições com relação a uma determinada descrição de um processo.
- Relacione os "conceitos da área do negócio" (classes no modelo de domínio) com fluxos de objetos nos modelos de atividade.

Modelação funcional com casos de utilização

Principais referências: TP2, TP3.

- Descrever o processo usado para identificar casos de utilização.
- Ler e criar diagramas de casos de utilização.
- Rever modelos de casos de utilização existentes para detectar problemas semânticos e sintáticos.
- Descrever os elementos essenciais de uma especificação de caso de uso.
- Explicar o uso complementar de diagramas de casos de utilização, diagramas de atividades e narrativas de casos de utilização.
- Explicar o sentido da expressão "desenvolvimento orientado por casos de utilização".
- Explicar os seis "Princípios para a adoção de casos de utilização" propostos por Ivar Jacobson (com relação ao "Use Cases 2.0")
- Compreender a relação entre requisitos e casos de utilização
- Identificar as disciplinas e atividades relacionadas aos requisitos no OpenUP

Modelação estrutural

Principais referências: TP05, TP06, TP10.

- Distinguir entre a análise de sistemas baseada num aabrodagem algorítmica *top-down* e baseada nos coneitos do domínio do problema.
- Justifique o uso de modelos estruturais na especificação de sistemas.
- Explicar a relação entre os diagramas de classe e de objetos.
- Rever um modelo de classes quanto a problemas de sintaxe e semânticos, considerando uma descrição do um problema de aplicação.
- Descreva os tipos e funções das diferentes associações no diagrama de classes.
- Identifique o uso adequado da associação, composição e agregação para modelar a relação entre objetos.
- Identifique o uso adequado de classes de associação.

Modelação de comportamento

Principais referências: TP08, TP09.

- Explique o papel da modelagem de comportamento no SDLC
- Entenda as regras e diretrizes de estilo para diagramas de seqüência, comunicação e estado
- Entenda a complementaridade entre diagramas de sequência e comunicação Diagramas de
- sequência de mapa em código orientado a objeto e reverso .
- Analise criticamente os modelos de diagramas de sequência existentes para descrever a cooperação entre dispositivos ou entidades de software.

Modelos no desenho e implementação

Vistas de arquitetura

Principais referências: TP10.

- Explicar as atividades associadas ao desenvolvimento de arquitetura de software.
- Identifique os elementos abstratos de uma arquitetura de software
- Identifique as camadas e partições numa arquitetura de software por camadas.
- Rever criticamente um diagrama de pacotes existente para ilustrar uma arquitetura lógica
- Rever criticamente um diagrama de componente existente para descrever as partes tangíveis do software
- Analise criticamente um diagrama de implementação existente para descrever a instalação de um sistema

Classes e desenho de métodos (perspectiva do programador)

Referências principais: TP06, TP08

- Explicar os princípios de baixo acoplamento e alta coesão no desenho por objetos.
- Enumerar e descrever os princípios do GRASP (Larman)
- Explicar as implicações no código da navegabilidade modelada no diagrama de classes.
- Construa um diagrama de classes e um diagrama de sequência considerando um código Java.

Práticas selecionadas na construção do software

Garantia de qualidade

Principais referências: TP12.

- Identifique as atividades de validação e verificação incluídas no SDLC
- Descreva quais são as camadas da pirâmide de teste
- Descreva o assunto/objetivo dos testes de unidade, integração, sistema e de aceitação
- Explique o ciclo de vida do TDD
- Descreva as abordagens “debug-later” e “test-driven”, de acrodo com J. Grenning.
- Explique como é que as atividades de garantia de qualidade (QA) são inseridas no processo de desenvolvimento, numa abordagem clássica e nos métodos ágeis.
- O que é o “V-model”?
- Relacione os critérios de aceitação da história (*user-story*) com o teste *Agile*.

Abordagens complementares

Histórias e métodos ágeis

Principais referências: TP13.

- Defina histórias (*user stories*) e dê exemplos.
- Explique a metáfora do "*post-it*" (para planeamento e seguimento) comum em projetos ágeis.
- Identifique os elementos-chave de uma "persona".
- Compare histórias e casos de utilização em relação a pontos comuns e diferenças.
- Compare "Persona" com Ator com respeito a semelhanças e diferenças.
- O que é a pontuação de uma história e como é que é determinada?
- Descreva o conceito de velocidade da equipa (como usado no PivotalTracker e SCRUM).
- Discutir se os casos de utilização e as histórias são abordagens redundantes ou complementares (quando seguir cad auma das abordagem? Em que condições? ...)

A metodologia SCRUM

Principais referências: palestra convidada 6/6.

- Explique o objetivo ca "Daily Scrum meeting"
- Realcione os conceitos de *sprint* e iteração e discuta a sua duração esperada.
- Explique a método de pontuação das histórias (e critérios aplicados)
- Relacione as práticas previstas no SCRUM e os princípios do "Agile Manifest": em que medida estão alinhados?

Recursos adicionais

Veja também:

- Exemplos de Exames anteriores

Dúvidas e dúvidas:

- Utilize preferencialmente o canal #mas_lei (https://detiuaveiro.slack.com).