01:

Identify three main axes in digital transformation:

User experience , processos operacionais e business model

Explain what the Systems Development Cycle is:

Processo de determinar como um sistema de informação pode ser criado. Tem quatro fases principais – planeamento, análise, design e implementação. Cada fase tem uma sequência de ações e no final de cada fase há entregas do que foi feito. O membro “principal” da equipa é o analista, que analisa a situação de negócio e identifica oportunidades de melhoria – que preparará para implementação.

02:

Justify the use of models in systems engineering

Modelos facilitam a visualização do sistema high level. Para além disso, especificam e documentam a estrutura e o comportamento do sistema antes da implementação, o que tornará a implementação em si mais smooth. Assim, também servem de referência para orientar a construção, também documentando as decisões feitas ao longo do processo de design.

Enumerate advantages of visual models

Ao criar modelos visuais, a comunicação tornar-se-á mais clara e sucinta. Para além disso, torna o desenho e a implementação coerentes. Além disso, tendo modelos visuais, ter-se-á um nível de detalhe apropriado para cada fase. Não só, certas ferramentas tornam a construção um processo automático através de modelos.

Identify the main diagrams in UML and their modeling viewpoint

Diagrama de use case - organiza as funcionalidades do sistema em episódios de utilização

Diagrama de atividade – explica procedimentos do domínio

Modelo de domínio – usa classes para representar os conceitos do problema, tornando as ligações entre eles mais claras

Diagrama de estados – análogo ao diagrama de atividade, mas com foco no estado de um componente

02b:

Distinguish structured activities from actions; control flows from data flows

Ações são algo que é feito ao sistema, enquanto que atividades estruturadas são, habitualmente, reações a ações, sendo intrínsecas a este. Control flows diferem de data flows por natureza – control flows ditam como algo vai acontecer e data flows apenas ditam fluxos de informação.

Identify when to use Activity modeling

Como diagramas de atividade mostram o fluxo de ações e dados, são úteis para modelar fluxos de trabalho/processos de negócio e processos computacionais/algoritmos. Também são úteis na descrição de sequências de interações entre atores num caso de utilização.

03:

Identificar atividades comuns a todos os projetos (ciclo de vida)

Identificação do valor que o sistema gerará para a organização, pedido do sistema e a análise de viabilidade são apresentadas às entidades que dão o “okay” a novos projetos, criação de plano de trabalho, análise dos sistemas existentes, recolha de requisitos, conceitualização de uma solução, definição de estratégia de desenvolvimento, desenho de arquitetura do sistema, modelo de dados e programas, implementação do sistema, instalação e transição para produção e, finalmente, execução do plano de suporte.

Distinguir projetos sequenciais de projetos evolutivos

Em projetos sequenciais, todas as atividades do processo são planeadas com antecedência e o progresso do projeto é comparado com o plano original, enquanto que num projeto evolutivo o planeamento é incremental, sendo possível adaptar caso os requisitos do cliente mudem.

Descrever a estrutura do Unified Process (fases, objetivos, iterações)

Inception – iteração curta que produz o documento relativo à visão e um business case inicial. Também reduz o risco, identificando os requirements chave

Elaboration – faz “sacrifícios” no que toca a riscos de business e produção de valor. Também produz e valida a arquitetura.

Construction - construção do que não foi feito na elaboração, criando um sistema operacional

Transition – validação final e deplyment

Mapear disciplinas técnicas nas fases do OpenUP

Na inception, as duas disciplinas técnicas com maior foco são business modelling e requirements. Na elaboração, o foco é colocado nos requirements e na análise e design. Na construção, é na implementação e configuração e management de alterações. Estas mesmas fases mantém o foco na transição.

04:

>write in the form of subject-verb-direct object

>make sure it’s clear who the initiator of the step is

>write from an independent observer’s perspective

>write at about the same level of abstraction

>ensure the use case has a sensible set of steps

>apply the KISS principle liberally (keep it simple, stupid)

>write repeating instructions after the set of steps to be repeated

What happens? Then what? What if one of the steps fails/doesn’t go that way?

05:

Distinguish functional and non-functional requirements

Requisitos funcionais são serviços e funções que o sistema tem de ser capaz de fazer, enquanto que requisitos não funcionais são restrições/compromissos no software em si.

Enumerate requirement gathering techniques and argument when to use each one

Entrevistas com o cliente são um bom método de encontrar requirements surface-level, numa fase inicial do projeto será bom estabelecer com o cliente quais são as necessidades base. Questionários com utilizadores futuros também dão origem a requirements surface-level ao nível funcional. Análise de documentos será útil para não cometer os erros que outros já cometeram, tal como na observação.

Describe requirements documentation techniques

Requisitos são, por norma, documentados em tabelas e são bem formados, isto é, podem ser verificados mensuravelmente.

Understand the relation between requirements and use cases

Um use case irá, por norma, assentar nos requisitos já estabelecidos, ou, na criação dele, irá tornar mais claros alguns requisitos que ainda não tinham sido considerados.

06: