**Análise de Sistemas – Exame Teórico 2022**

**A) O que é que está incluído no SDLC?**

-O ciclo de vida de um sistema de informação compreende todas os processos que ocorrem desde o nascimento da necessidade de um sistema ate ao surgimento de outro que o substitui.

-Um sistema de informação inclui todas as pessoas, maquinas e/ou métodos envolvidos na coleta, processamento e transmissão de informação.

-Normalmente, as funções identificadas na implementação de um sistema de informações são o gerente de projetos, técnicos na área de tecnologia analistas e usuários.

**Objetivos de um sistema de informação:**

-Definir as tarefas a serem executadas e a ordem em que devem ser executadas;

-Garantir a consistência com o restante dos sistemas de informação de organização;

-Fornecer pontos de controle para gerenciamento de projetos;

As seis fases do ciclo de vida de um sistema de informação:

-**1ª Passo** “Planeamento”: compreensão do porquê de um sistema de informação deve ser construído e determinar como a equipa do projeto irá construí-lo. O valor do sistema para a organização é identificado, breve resumo da necessidade de negócio e explica como o novo sistema ira suportar isto;

-**2º Passo** “Análise”: esclarece quem irá utilizar o sistema, o que é que faz o sistema, e onde e como será utlizado; identifica melhorias a fazer perante outros sistemas; Estudo do domínio, Levantamento de requisitos; Conceito para solução;

-**3º Passo** “Desenho”: decide como o novo sistema será construído, software, hardware, infraestruturas de rede, interface, … Escolha de tecnologias e tática para equipa de desenvolvimento; Estratégia de desenvolvimento, conceção da arquitetura, conceção do modelo de dados, desenho das entidades de software e seleção de frameworks;

-**4º Passo** “Implementação”: o sistema é de facto construído, com escrita de código, integração de sistemas, desenvolvimento das bases de dados, testes ao software; Implementação de sistemas, instalação e transição, plano de suporte(gestão de modificações);

Para alem disto ainda temos uma fase de manutenção.

O SDLC é concretizado usando um processo de software sistemático.

O **processo de software** é um guião para as atividades e tarefas que são necessárias para construir software de qualidade. (necessitamos de um processo explicito, pois, as falhas são frequentes e criar Sis. Complexos não é intuitivo)

O **analista de sistemas** analisa a situação de negócio, identifica oportunidades de melhorias e projeta um sistema de informação para implementá-las. Ser analista ade sistemas é um dos trabalhos mais desafiantes na eng. de software. O principal objetivo do analista não é criar os sistemas” topo de gama” mas é criar valor para a organização;

**Na abordagem ágil** a garantia de qualidade tem de ser aplicada em cada ciclo.

**Processos de software e o Unfield Process/OpenUp**

-O OpenUp é um projeto open source que define um framework de processo de desenvolvimento de software. Possui uma abordagem iterativa e incremental dentro de um ciclo de vida. Pode ser estendido para direcionar uma grande variedade de tipos de projeto pois esta dentro dos princípios do desenvolvimento de software ágil e é baseado em casos de uso.

-O UP oferece uma abordagem ao SDLC concebida como uma matriz cruzando diferentes disciplinas com iterações no projeto. A analise de requisitos é realizada principalmente no inicio do projeto(básicos) mas também durante as iterações.

-O **OpenUP** estrutura o ciclo de vida do projeto em 4 fases:

**Conceção**- os stakeholders e os membros da equipa colaboram para determinar os objetivos do projeto deve ou não continuar.

**Elaboração**- os riscos arquiteturais significantes são tratados;

**Construção**- foco no detalhamento dos requisitos, no desenho, na implementação e nos testes da maior parte do software.

**Transição**- focada na transição do software para o ambiente do cliente e na concordância de que o desenvolvimento do produto está completo.

Os **Milestones(**é o género de um marco) determinam formalmente o final de uma fase e funcionam como uma verificação para poder prosseguir para a fase seguinte.

O OpenUP descreve como encontrar os requisitos com uma combinação de casos de uso, e dá-os ao programador para que este faça testes em função dos mesmos casos de uso;

**Modelação visual e UML**

.

.

.

**B) Compreender as necessidades de negócio**

-Determinação de requisitos tem como objetivo converter pedidos de negócio em requisitos detalhados que possam ser utilizados como inputs para a criação de modelos.

-Um requisito é uma declaração do que o sistema deve fazer ou uma característica que deve ter, que mais tarde evoluirá para uma descrição técnica de como o sistema será implementado; desenvolver um sistema em conformidade com os requisitos estabelecidos.

-**Requisitos Funcionais**: refere-se a comportamento, coisas/tarefas que o sistema é capaz de realizar; O agrupamento dos RF depende doo problema especifico;

-**Requisitos não funcionais:** é uma qualidade ou propriedade do sistema; ex: As pesquisas de utentes devem retornar os resultados em menos de um segundo; Categorias comuns: Usabilidade, Desempenho, Segurança, Restrições operacionais;

Técnicas de recolha de requisitos: entrevistas, workshops, questionários, observação.

Técnicas alternativas: analise da área, user stories, mapas de conceitos;

Consoante a ISSO-IEEE 29148 um requisito bem formulado tem de ser satisfeita ou detida apor um sistema para resolver um problema, pode ser verificada, qualificado por mensuráveis e limitado por restrições e define o comportamento ou a capacidade do sistema.

Segundo os critérios de S.M.A.R.T os requisitos devem ser escritos com base na mnemónica: específicos, mensuráveis, alcançáveis, relevantes e sensível ao tempo.

Casos de utilização engloba uma sequência de ações que um sistema executa e produzem um resultado com valor para algum ator em particular; Implica foco na compreensão do que os atores consideram um resultado relevante pois visa os problemas que querem resolver.

**Requisitos centrados no produto**- pensamento focado em caraterísticas técnicas do que seria um bom produto;

**Requisitos centrados no utilizador**- captar requisitos ao contar historias; Exemplo: User Storie- O medico insere o nº utente do SNS e confirma a pesquisa; Requisito- O sistema deve permitir a pesquisa de cheques por numero de utente do SNS.

SRS-Especificação de requisitos de software;

**Wieger’s Proposal:**

Não se tem de escrever o SRS para o produto inteiro antes de se iniciar o desenvolvimento; necessário definir os requisitos para cada incremento antes de o contruir.

Introdução, descrição geral, recursos do sistema, requisitos data, requisitos da interface externa, atributos de qualidade, requisitos de internacionalização e localização, outros requisitos;

Abordagem para analise de sistemas:

-Dominio- novos processos de trabalho, modelo de conceitos do domínio;

-Requisitos- Pontos de partida: use cases;

**Requisitos evolutivos:**

-Plan driven, “waterfall” approach: estabilizar os requisitos na primeira fase de um projeto;

-Agile, evolutionary approach: suporte de algum tipo de abordagem sistemática para encontrar e acompanhar a evolução dos requisitos no sistema;

**Casos de uso:**

Iniciados por um ator primário;

Ator primário- Solicita o sistema para resolver um problema.

Ator secundário- fornece serviços ou informação para algum cenário no caso de uso; podem ser sistemas externos ou papeis de pessoas que não são utilizadores.

A **arquitetura de software** de um sistema é o conjunto de estruturas necessárias para raciocinar sobre o sistema. Essas estruturas compreendem elementos de software, relações entre eles e propriedades de ambos.

-**Estrutura de componentes e conetores** focam-se na forma como os elementos interagem uns com os outros em tempo de execução para realizar as funções do sistema.

-A **estruturas do tipo modulo** dividem os sistemas em unidades de implementação. Os módulos mostram como um sistema pode ser dividido como um conjunto de unidades de código ou de dados, que devem ser implementadas ou adquiridas (relação de “uses”).

-As **estruturas de alocação** estabelecem o mapeamento das estruturas de software nas estruturas de não software do sistema. Responde a questões de em que processador cada elemento de software é executado, em que diretório é armazenado.

Um **requisito de arquitetura** significativo (ASR) é um requisito que terá um efeito profundo na arquitetura, ou seja, a arquitetura pode muito bem ser substancialmente diferente na ausência de tal requisito.

**Arquitetura por camadas** é a divisão modular da solução de software em camadas; as camadas são sobrepostas, cada camada tem uma especialização, as camadas de cima pedem serviços as camadas de baixo e não se podem saltar camadas.

**Desenvolvimento iterativo e incremental:**

Iterativo- atividade em que a criação de software é realizada por meio de progressos sucessivos, apresentado por etapas ate que o produto final seja alcançado;

Incremental- aquele em que o software é entregue aos pedaços, já são entregues na sua versão final, fazendo com que haja uma concentração de esforços em determinadas partes para que estejam a funcionar.

Existe uma repetição de das etapas do processo no desenvolvimento iterativo do software ate o resultado ser obtido; já no desenvolvimento incremental novos incrementos são adicionados ao longo do período de construção.

A criação do iterativo e incremental está nas bases das metodologias ágeis, como a SCRUM. A ideia é que a criação de software seja dividida por vários ciclos curtos em que as funcionalidades são introduzidas, feedbacks coletados e requisitos revistos. Para atingir um maior nível de satisfação do cliente e garantir o sucesso do resultado final.

**Metodologias ágeis** foram criadas para tornar o desenvolvimento de software mais rápido e produtivo, é um conjunto de princípios com foco na colaboração e praticas são modificadas conforme o contexto do projeto. Tem como base a realização de reuniões constantes de acompanhamento de resultados, interações com o cliente e incrementos. De modo, a deixar o trabalho mais colaborativo e rápido, maximizando o sucesso do projeto.

O **Scrum** é um método ágil de desenvolvimento de projetos. Focado na transparência, ele reduz o tamanho das etapas e cria um fluxo de trabalho que permite a entrega mais rápida de resultados e mais agilidade para os times.

**Frases para comentar**

Pergunta: “As pessoas falam frequentemente de “recolha de requisitos”. Esta frase sugere que os requisitos estão por aí á espera de serem recolhidos como flores, ou de serem retirados do cérebro dos utilizadores pelo analista. Eu prefiro o termo elicitação de requisitos a recolha de requisitos. “

Resposta: O levantamento de requisitos de software é um processo que serve para capturar as necessidades do cliente antes de projetar o desenvolvimento. Assim, os problemas selecionados pelo sistema serão problemas reais e não apenas imaginados pela equipa de desenvolvimento. A elicitação de requisitos passa por perceber as necessidades do sistema e as características que esse sistema deve ter. Nesta fase o analista deve andar em busca de funcionalidades que o sistema deve ter, restrições e usabilidade de software de modo a solucionar estes problemas para maximizar o sucesso do novo software.