

Candidato: Dario Martins Bueno Leandro

1) EXPLIQUE COM SUAS PALAVRAS O QUE É DOMAIN DRIVEN DESIGN E SUA IMPORTÂNCIA NA ESTRATÉGIA DE DESENVOLVIMENTO DE SOFTWARE.

R:

O DDD é voltado para domínios complexos, e depende da quebra de barreiras de comunicação entre especialistas de negócio e especialistas técnicos, além do engajamento do time técnico em programar de forma que o código reflita o domínio.

Para quem já leu o livro do Eric Evans (Domain-Driven Design) ele aborda a preocupação no entendimento do negócio pois a partir desse momento que podemos implementar o DDD em um projeto.

Para ter um software que atenda perfeitamente a um determinado domínio, é necessário que se estabeleça, em primeiro lugar, uma Linguagem Ubíqua (Linguagem comum, com termos bem definidos, que fazem parte do domínio do negócio e que são usados por todas as pessoas que fazem parte do processo de desenvolvimento de software).

Em termos de estratégia a parte mais importante (e difícil) de Domain Driven Design é quando começamos a lidar com sistemas complexos. Como deve ser a interação entre sistemas que interagem entre si? Como dividir nosso trabalho, de maneira que foquemos nossos esforços naquilo que tem maior valor para o negócio? Como fazer a comunicação entre os times que desenvolvem esses sistemas?

Temos alguns padrões que nos ajudam a dividir nosso software em várias partes, que chamamos de contextos. Cada **Contexto Delimitado** deve estar bem claro para todos que estão envolvidos no processo de desenvolvimento. A fronteira entre contextos deve ser clara para todos, ou seja, todo mundo deve saber a qual contexto um determinado pedaço de código pertence.

Eu lembro que no livro do Eric Evans ele não tinha criado patterns específico apenas fez o uso correto deles criando uma abordagem de modelagem tática que incluem os seguintes componentes: Aggregate Object, Domain Model, Value Object, Factory, Domain Service, Application Service, Repository, External Service.

2) EXPLIQUE COM SUAS PALAVRAS O QUE É E COMO FUNCIONA UMA ARQUITETURA BASEADA EM MICROSERVICES. EXPLIQUE GANHOS COM ESTE MODELO E DESAFIOS EM SUA IMPLEMENTAÇÃO.

R:

Os micros serviços são uma arquitetura e uma abordagem para escrever programas de software. Com eles, as aplicações são desmembradas em componentes mínimos e independentes. Diferentemente da abordagem tradicional monolítica em que toda a aplicação é criada como um único bloco, os micros serviços são componentes separados que trabalham juntos para realizar as mesmas tarefas. Cada um dos componentes ou processos é um micro serviço. Essa abordagem de desenvolvimento de software valoriza a granularidade, a leveza e a capacidade de compartilhar processos semelhantes entre várias aplicações. Trata-se de um componente indispensável para a otimização do desenvolvimento de aplicações para um modelo nativo em nuvem.

Com relação aos ganhos está na rapidez com que a solução permite entregar software com qualidade. Isso é possível com a adoção dos micros serviços, mas não é só isso. Não basta apenas desmembrar as aplicações em micros serviços. Também é necessário gerenciar esses micros serviços e orquestrá-los, além de processar os dados que são gerados e modificados por eles.

Com relação aos desafios eu acredito que está no entendimento do negócio, identificar os domains experts para guiá-lo no desenvolvimento.

Extrair linguagens ubíqua onde estão todos os termos que fazem parte das conversas diárias entre especialistas de negócio e times de desenvolvimento. Todos devem usar os mesmos termos tanto na linguagem falada quanto no código. Isso significa que, se durante uma conversa com um cliente do sistema de cobrança, por exemplo, ele disser: “Temos que emitir a fatura para o cliente antes da data limite”, vamos ter no nosso código alguma coisa do tipo:

- Uma classe para a entidade Cliente;
- Uma classe para a entidade Fatura;
- Algum serviço que tenha um método emitir;
- Algum atributo com o nome de data limite.

Elaborar uma modelagem estratégica onde permitirá ter uma visão de entendimento dos negócios em partes menores devido ao entendimento da linguagem ubíqua.

3) EXPLIQUE QUAL A DIFERENÇA ENTRE COMUNICAÇÃO SÍNCRONA E ASSÍNCRONA E QUAL O MELHOR CENÁRIO PARA UTILIZAR UMA OU OUTRA

R:

#### **Comunicação Síncrona**

A comunicação síncrona é quando temos a oportunidade de falar com alguma pessoa de maneira direta. Ou seja, o emissor passa a mensagem e o receptor responde, sendo que o diálogo ocorre naquele momento.

Dessa forma, as mensagens emitidas por uma pessoa são imediatamente recebidas e respondidas por outras pessoas.

Como exemplo de ferramentas de comunicação síncrona podemos citar o contato via telefone, reuniões, aulas de ensino presencial etc.

#### **Comunicação Assíncrona**

Quando se fala em rotinas assíncronas está se tratando de uma forma de usar o processamento paralelo onde não é necessário aguardar o término de uma rotina para executar outra. Este tipo de rotina é usado quando não é necessário aguardar o seu término ou quando o resultado não é necessário para que se possa continuar usando o software.

Exemplos bem práticos de rotinas assíncronas são o envio de e-mail, um envio de solicitação de processamento para um banco de dados (como uma consulta, um relatório, um cálculo que possa ser executado enquanto o usuário esteja fazendo outras tarefas).