

Revisão e refino do modelo de domínio (modelagem de classes de análise)

Conceitos, processo de refinamento, exemplos práticos e aplicação

1. O que é o modelo de domínio?

O modelo de domínio é uma representação visual dos principais conceitos do negócio que o software vai tratar, junto com seus atributos, comportamentos e relacionamentos. Diferente de um diagrama de classes de projeto (design), que já pensa em detalhes técnicos como métodos de acesso a banco de dados ou padrões de arquitetura, o modelo de domínio foca exclusivamente em entender o problema do mundo real, sendo por isso chamado também de **diagrama de classes de análise**.

Ele é normalmente representado usando a notação UML (Unified Modeling Language), com classes representando conceitos importantes do negócio (como Cliente, Pedido ou Produto), atributos representando suas características e associações representando como esses conceitos se relacionam entre si.

2. Por que revisar e refinar o modelo de domínio?

A primeira versão de um modelo de domínio, criada logo após o levantamento inicial de requisitos, raramente está completa ou totalmente correta. Conforme a equipe aprofunda a análise, conversa mais com especialistas do negócio e descobre casos de uso mais complexos, é natural que o modelo precise ser revisado e refinado. Esse processo é essencial porque:

- Evita que erros de entendimento do negócio se propaguem para as fases de projeto e desenvolvimento, onde seriam muito mais caros de corrigir.
- Garante que o vocabulário usado no modelo reflita exatamente os termos utilizados pelos especialistas do domínio, evitando ambiguidades.
- Identifica classes desnecessárias (que na verdade são apenas atributos), classes ausentes (conceitos importantes que foram esquecidos) e relacionamentos incorretos ou incompletos.
- Serve de base sólida e confiável para as próximas etapas, como o desenho de casos de uso detalhados e o diagrama de classes de projeto.

3. Etapas do processo de revisão e refino

3.1 Identificação inicial das classes candidatas

A primeira versão do modelo costuma surgir da análise dos substantivos presentes nos requisitos, histórias de usuário ou entrevistas com o cliente. Nem todo substantivo vira uma classe: é preciso

avaliar quais representam de fato conceitos relevantes e persistentes para o negócio.

Exemplo prático: em um sistema de biblioteca, a partir da frase "o leitor empresta um livro que pertence a uma categoria e é escrito por um autor", surgem as classes candidatas Leitor, Livro, Categoria, Autor e Empréstimo.

3.2 Eliminação de classes redundantes ou irrelevantes

Nesta etapa, a equipe revisa cada classe candidata e remove aquelas que na verdade são apenas atributos de outra classe, que são sinônimos de conceitos já existentes ou que estão fora do escopo do sistema.

Exemplo prático: ao revisar o modelo da biblioteca, a equipe percebe que "Data de Devolução" havia sido erroneamente modelada como uma classe separada, quando na verdade é apenas um atributo da classe Empréstimo.

3.3 Refinamento de atributos e comportamentos

Cada classe mantida é detalhada com seus atributos relevantes para o negócio (sem incluir ainda detalhes técnicos de implementação) e, quando aplicável, comportamentos de alto nível que representam regras importantes do domínio.

Exemplo prático: a classe Empréstimo é refinada para incluir os atributos data de início, data prevista de devolução e status (ativo, devolvido, atrasado), além do comportamento "verificar se está atrasado", que representa uma regra de negócio importante para a biblioteca.

3.4 Revisão dos relacionamentos e cardinalidades

As associações entre classes são revisadas quanto ao tipo de relacionamento (associação simples, agregação ou composição) e à cardinalidade (quantas instâncias de uma classe se relacionam com quantas instâncias de outra), corrigindo interpretações equivocadas da primeira versão.

Exemplo prático: inicialmente modelado como um relacionamento "um para um" entre Livro e Autor, o modelo é corrigido para "muitos para muitos", já que um livro pode ter mais de um autor e um autor pode ter escrito vários livros.

3.5 Validação com especialistas do domínio

O modelo refinado é apresentado a especialistas do negócio (bibliotecários, no exemplo usado) para confirmar se reflete corretamente a realidade, corrigindo eventuais mal-entendidos antes de avançar para as próximas fases do projeto.

Exemplo prático: ao validar o modelo com a equipe da biblioteca, descobre-se que também é preciso representar reservas de livros já emprestados, levando à inclusão de uma nova classe, Reserva, que não havia sido identificada anteriormente.

4. Exemplo simplificado de notação UML

Um trecho do modelo de domínio da biblioteca, já refinado, pode ser representado textualmente da seguinte forma (equivalente a um diagrama de classes UML simplificado):

```
Leitor (nome, matrícula) 1 ---- * Empréstimo (data_início, data_devolução, status) * ---- 1 Livro (título, ano_publicação)
Livro * ---- * Autor (nome)
Livro 1 ---- 1 Categoria (nome)
Leitor 1 ---- * Reserva (data_solicitação) * ---- 1 Livro
```

5. Resumo visual do processo de refino

Etapa	Objetivo
Identificação de classes candidatas	Levantar conceitos a partir dos requisitos
Eliminação de redundâncias	Remover classes desnecessárias ou duplicadas
Refinamento de atributos	Detalhar características relevantes de cada classe
Revisão de relacionamentos	Corrigir associações e cardinalidades
Validação com especialistas	Confirmar que o modelo reflete o negócio real

6. Boas práticas na revisão do modelo de domínio

- Usar sempre os termos do vocabulário do negócio (linguagem ubíqua), evitando termos técnicos que não fazem sentido para os especialistas do domínio.
- Manter o modelo simples na fase de análise, sem incluir detalhes de implementação como tipos de dados específicos de banco de dados ou padrões de projeto.
- Revisar o modelo em conjunto com toda a equipe (analistas, desenvolvedores e representantes do negócio), e não apenas individualmente.
- Repetir o ciclo de refinamento sempre que novos requisitos ou casos de uso forem descobertos ao longo do projeto, especialmente em contextos ágeis.

7. Aplicação prática no dia a dia dos projetos

A revisão e o refino do modelo de domínio são aplicados em diferentes momentos e por diferentes profissionais:

- **Analistas de sistemas** utilizam o modelo refinado como base para escrever casos de uso e especificações mais detalhadas e consistentes.
- **Arquitetos e desenvolvedores** partem do modelo de domínio validado para criar o diagrama de classes de projeto, já incluindo decisões técnicas de implementação.
- **Product Owners**, em equipes ágeis, usam o modelo de domínio para garantir que toda a equipe compartilhe o mesmo entendimento sobre os conceitos centrais do produto, mesmo sem documentação extensa.

- **Equipes de qualidade (QA)** utilizam o modelo para elaborar casos de teste que cubram corretamente as regras de negócio representadas nos relacionamentos e comportamentos das classes.

8. Considerações finais

A revisão e o refino do modelo de domínio são etapas fundamentais para garantir que o software esteja alicerçado em uma compreensão sólida e validada do negócio que ele pretende apoiar. Investir tempo nessa atividade reduz retrabalho nas fases seguintes, melhora a comunicação entre equipe técnica e especialistas do negócio, e aumenta as chances de que o sistema final realmente atenda às necessidades reais dos usuários.