Capítulo 9

Casos de Uso

Os casos de uso são uma técnica para captar os requisitos funcionais de um sistema. Eles servem para descrever as interações típicas entre os usuários de um sistema e o próprio sistema, fornecendo uma narrativa sobre como o sistema é utilizado.

Em vez de descrever os casos de uso de início, acho mais fácil rodeá-los e começar descrevendo cenários. Um **cenário** é uma seqüência de passos que descreve uma interação entre um usuário e um sistema. Assim, se tivermos uma loja *on-line* baseada na Web (loja virtual), podemos ter um cenário de Compra de um Produto que diria:

> *O cliente navega no catálogo de itens e adiciona os itens desejados à sua cesta de compras. Quando o cliente deseja pagar, descreve o endereço de entrega, fornece as informações do cartão de crédito e confirma a venda. O sistema verifica a autorização do cartão de crédito e confirma a venda imediatamente e com um* e-mail *subseqüente.*

Esse cenário é uma alternativa que pode acontecer. No entanto, a autorização do cartão de crédito pode falhar, o que seria um outro cenário. Em um outro caso, você poderia ter um cliente regular de quem não precisa captar o endereço de entrega e as informações do cartão de crédito, o que seria um terceiro cenário.

Todos esses cenários são diferentes, embora semelhantes. A essência de sua similaridade é que, em todos, o usuário tem o mesmo objetivo: comprar um produto. Nem sempre ele tem sucesso, mas o objetivo permanece. Esse objetivo do usuário é a chave dos casos de uso: um **caso de uso** é um conjunto de cenários amarrados por um objetivo comum de usuário.

No jargão dos casos de uso, os usuários são referidos como atores. Um **ator** é um papel que um usuário desempenha com relação ao sistema. Os atores podem ser o cliente, representante de serviço ao cliente, gerente de vendas e analista de produto. Os atores realizam os casos de uso. Um único ator pode realizar muitos casos de uso; inversamente, um caso de uso pode ter vários atores executando-o. Normalmente, você tem muitos clientes; portanto, muitas pessoas podem ser o ator cliente. Além disso, uma pessoa pode atuar como mais de um ator, como um gerente de vendas que executa tarefas de representante de serviço ao cliente. Um ator não precisa ser um ser humano. Se o sistema realiza um serviço para outro sistema de computador, esse outro sistema é um ator.

Na realidade, *ator* não é o termo correto; *papel* seria muito melhor. Aparentemente, houve uma tradução errada do sueco; *ator* é o termo que a comunidade dos casos de uso utiliza.

Os casos de uso são reconhecidos como uma parte importante da UML. Entretanto, a surpresa é que, de muitas maneiras, a definição de casos de uso na UML é muito

rala. Nada na UML descreve como você deve capturar o conteúdo de um caso de uso. O que a UML descreve é um diagrama de casos de uso, que mostra como utilizar casos relacionados entre si. Mas praticamente todo o valor dos casos de uso reside no conteúdo e o diagrama é de valor bastante limitado.

CONTEÚDO DE UM CASO DE USO

Não existe nenhuma maneira padronizada para escrever o conteúdo de um caso de uso e diferentes formatos funcionam bem em diferentes casos. A Figura 9.1 mostra um estilo comum de uso. Você começa escolhendo um dos cenários como sendo o **cenário principal de sucesso (CPS)**. Dá início ao corpo do caso de uso escrevendo o cenário principal de sucesso como uma seqüência de passos numerados. Então, pega os outros cenários e os escreve como **extensões**, descrevendo-os em termos de variações em relação ao cenário principal de sucesso. As extensões podem ser bem-sucedidas – o usuário atinge o objetivo, como em 3a – ou falhas, como em 6a.

Cada caso de uso tem um ator principal, que pede ao sistema para que execute um serviço. O ator principal é aquele cujo objetivo o caso de uso está tentando satisfazer e, normalmente (mas nem sempre) é o iniciador do caso de uso. Podem existir outros atores com os quais o sistema se comunica enquanto executa o caso de uso. Eles são conhecidos como atores secundários.

Cada passo em um caso de uso é um elemento da interação entre um ator e o sistema. Cada passo deve ser uma declaração simples e mostrar claramente quem está executando o passo. O passo deve mostrar a intenção do ator e não os mecanismos do que o ator faz. Conseqüentemente, você não descreve a interface com o usuário no caso de uso. Na verdade, a escrita do caso de uso normalmente precede o projeto da interface com o usuário.

Uma extensão dentro do caso de uso nomeia uma condição que resulta em diferentes interações daquelas descritas no cenário principal de sucesso e informa quais são essas

Compra de um Produto

Nível do Objetivo: Nível do Mar

Cenário Principal de Sucesso:

1. O cliente navega pelo catálogo e seleciona itens para comprar
2. O cliente vai para o caixa
3. O cliente preenche o formulário da remessa (endereço de entrega; opção de entrega imediata ou em três dias)
4. O sistema apresenta a informação completa do faturamento, incluindo a remessa
5. O cliente preenche a informação de cartão de crédito
6. O sistema autoriza a compra
7. O sistema confirma imediatamente a venda
8. O sistema envia uma confirmação para o cliente por *e-mail*

Extensões:

3a: Cliente regular
.1: O sistema mostra a informação atual da remessa, a informação de preço e a informação de cobrança
.2: O cliente pode aceitar ou escrever por cima desses padrões, retornando ao CPS, no passo 6

6a. O sistema falha na autorização da compra a crédito
.1: O cliente pode inserir novamente a informação do cartão de crédito ou cancelar

Figura 9.1 Exemplo de texto de caso de uso.

diferenças. Inicie a extensão dando um nome ao passo em que a condição é detectada e forneça uma breve descrição da condição. Após a condição, coloque passos numerados, no mesmo estilo que o do cenário principal de sucesso.

Conclua esses passos descrevendo onde você volta para o cenário principal de sucesso, caso volte.

A estrutura do caso de uso é uma excelente maneira de criar alternativas ao cenário principal de sucesso. Para cada passo, pergunte: como isso poderia ser feito de uma forma diferente? E em particular, pergunte: o que poderia dar errado? Normalmente é melhor pensar em todas as condições de extensão primeiro, antes de se aprofundar na solução das conseqüências. Dessa maneira, você provavelmente considerará mais condições, o que se traduz em menos erros a serem capturados posteriormente.

Um passo complicado em um caso de uso pode ser um outro caso de uso. Em termos de UML, dizemos que o primeiro caso de uso **inclui** o segundo. Não existe nenhuma maneira padronizada para mostrar no texto um caso de uso incluído, mas acho que o sublinhado, que sugere um elo de hipertexto, funciona muito bem e, em muitas ferramentas, será realmente isso. Assim, na Figura 9.1, o primeiro passo inclui o caso de uso "navega pelo catálogo e seleciona itens para comprar".

Os casos de uso incluídos podem ser úteis em um passo complexo que congestionaria o cenário principal ou em passos que são repetidos em vários casos de uso. No entanto, não tente subdivir os casos de uso em sub-casos de uso e sub-sub-casos de uso, utilizando decomposição funcional. Tal decomposição é uma boa maneira de perder muito tempo.

Assim como acontece com os passos nos cenários, você pode adicionar algumas outras informações comuns a um caso de uso.

- Uma **pré-condição** descreve o que o sistema deve garantir como verdadeiro, antes de permitir que o caso de uso comece. Isso é útil para dizer aos programadores quais condições eles não precisam verificar no código.
- Uma **garantia** descreve o que o sistema irá assegurar no final do caso de uso. As garantias de sucesso se mantêm após um cenário bem-sucedido; as garantias mínimas se mantêm após qualquer cenário.
- Um **gatilho** especifica o evento que inicia o caso de uso.

Quando você estiver considerando a adição de elementos, seja cético. É melhor fazer muito pouco do que fazer demais. Além disso, faça o máximo para manter o caso de uso breve e fácil de ler. Descobri que os casos de uso longos e detalhados não são lidos, o que anula seu objetivo.

O volume de detalhes necessário em um caso de uso depende do risco nesse caso de uso. Freqüentemente, você precisa de detalhes apenas no início de alguns poucos casos de uso importantes; os outros podem ser considerados imediatamente antes de você implementá-los. Você não precisa escrever todos os detalhes; a comunicação verbal é freqüentemente muito eficaz, particularmente dentro de um ciclo iterativo em que as necessidades são rapidamente atendidas por meio da execução do código.

DIAGRAMAS DE CASOS DE USO

Conforme eu disse anteriormente, a UML nada diz sobre o conteúdo de um caso de uso, mas fornece um formato de diagrama para mostrá-lo, como se vê na Figura 9.2. Embora

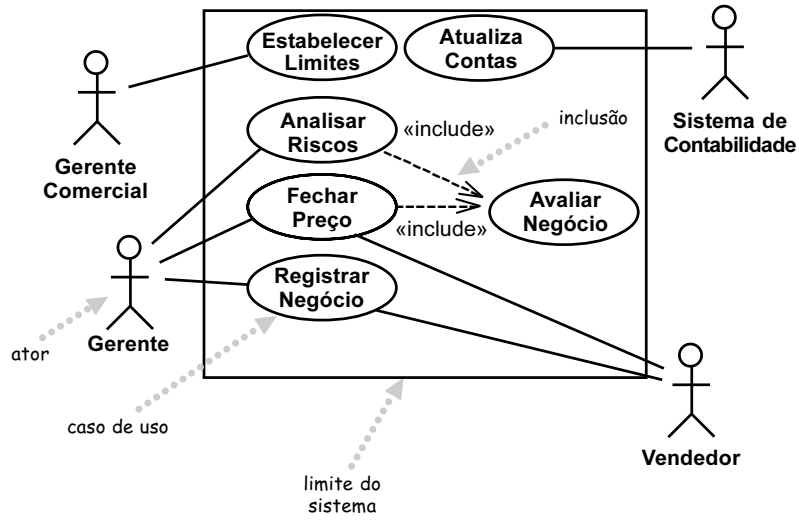

FIGURA 9.2 Diagrama de casos de uso.

o diagrama às vezes seja útil, ele não é obrigatório. Em seu trabalho com casos de uso, não se esmere muito no diagrama. Em vez disso, concentre-se no conteúdo textual dos casos de uso.

A melhor maneira de pensar um diagrama de caso de uso é como um sumário gráfico do conjunto de casos de uso. Ele também é semelhante ao diagrama de contexto usado nos métodos estruturados, pois mostra o limite do sistema e as interações com o mundo exterior. O diagrama de casos de uso mostra os atores, os casos de uso e os relacionamentos entre eles:

- Quais atores realizam quais casos de uso
- Quais casos de uso incluem outros casos de uso

A UML inclui outros relacionamentos entre os casos de uso, além da inclusão simples, como «extend». Sugiro que você os ignore. Tenho visto muitas situações em que as equipes podem ficar terrivelmente atrasadas ao usar diferentes relacionamentos de caso de uso e muita energia é desperdiçada. Em vez disso, concentre-se na descrição textual de um caso de uso; é aí que reside o valor real da técnica.

NÍVEIS DE CASOS DE USO

Um problema comum que pode acontecer com casos de uso é que, concentrando-se na interação entre um usuário e o sistema, você pode negligenciar situações nas quais uma mudança no processo do negócio pode ser a melhor maneira de lidar com o problema. Freqüentemente, as pessoas falam sobre casos de uso de sistema e casos de uso de negócio. Os termos não são precisos, mas o uso geral é que um **caso de uso de sistema** é uma interação com o *software*, enquanto um **caso de uso de negócio** examina como a aplicação responde ao cliente ou a um evento.

[Cockburn, use cases] sugere um esquema de níveis de casos de uso. Os casos de uso básicos estão "no nível do mar". Normalmente, os casos de uso no **nível do mar** representam uma interação distinta entre um ator principal e o sistema. Tais casos de uso transmitirão algo de valor para o ator principal e, normalmente, este levará de alguns minutos a meia hora para terminar. Os casos de uso existentes apenas porque foram incluídos pelos casos de uso de nível do mar estão **em nível de peixe**. Os casos de uso de nível mais alto (**em nível de pássaro**) mostram como os casos de uso de nível do mar se encaixam nas interações do negócio mais amplas. Os casos de uso em nível de pássaro normalmente são casos de uso de negócio, enquanto os casos de níveis do mar e de peixe são casos de uso de sistema. Você deverá ter a maioria de seus casos de uso em nível do mar. Eu prefiro indicar o nível no início do caso de uso, como na Figura 9.1.

CASOS DE USO E FUNCIONALIDADES (OU HISTÓRIAS)

Muitas estratégias utilizam as funcionalidades de um sistema – a Extreme Programming os chama de histórias de usuário – para ajudar a descrever requisitos. Uma questão comum é como funcionalidades e casos de uso se correlacionam.

As funcionalidades constituem uma boa maneira de repartir um sistema para planejar um projeto iterativo, pelo qual cada iteração implementa várias funcionalidades. Os casos de uso fornecem uma narrativa de como os atores utilizam o sistema. Então, embora as duas técnicas descrevam requisitos, seus propósitos são diferentes.

Embora você possa passar diretamente à descrição das funcionalidades, muitas pessoas acham interessante desenvolver primeiro os casos de uso e depois gerar uma lista de funcionalidades. Um recurso pode ser um caso de uso inteiro, um cenário em um caso de uso, um passo em um caso de uso ou algum comportamento variante, como a adição de um outro método de depreciação para suas avaliações de bens, que não apareça em uma narrativa de caso de uso. Normalmente, os recursos acabam sendo mais refinados do que os casos de uso.

QUANDO UTILIZAR CASOS DE USO

Os casos de uso são uma ferramenta valiosa para ajudar no entendimento dos requisitos funcionais de um sistema. Uma primeira passagem nos casos de uso deve ser feita no início. Versões mais detalhadas dos casos de uso devem ser elaboradas apenas antes do desenvolvimento desse caso de uso.

É importante lembrar que casos de uso representam uma visão *externa* do sistema. Como tal, não espere quaisquer correlações entre eles e as classes dentro do sistema.

Quanto mais eu observo os casos de uso, menos valioso parece ser o diagrama de casos de uso. Com os casos de uso, você concentra sua energia no texto e não no diagrama. A despeito do fato de que a UML nada tem a dizer sobre o texto do caso de uso, é esse texto que contém todo o valor da técnica.

Um grande perigo dos casos de uso é que as pessoas os tornam complicados demais e não conseguem prosseguir. Normalmente, você terá menos problemas fazendo pouco do que fazendo demais. Umas duas páginas por caso de uso está bom, para a maioria das situações. Se você tiver muito pouco, pelo menos terá um documento curto e legível,

que será um ponto de partida para perguntas. Se você tiver demais, dificilmente alguém o lerá e o entenderá.

ONDE ENCONTRAR MAIS INFORMAÇÕES

Os casos de uso foram popularizados originalmente por Ivar Jacobson [Jacobson, OOSE].

Embora os casos de uso já existam há algum tempo, havia pouca padronização para seu uso. A UML nada diz sobre o importante conteúdo de um caso de uso e tem padronizado apenas os diagramas, muito menos importantes. Como resultado, você pode encontrar uma variedade de opiniões divergentes a respeito dos casos de uso.

Nos últimos anos, entretanto, [Cockburn, use cases] tornou-se o livro padrão sobre o assunto. Neste capítulo, segui a terminologia e as recomendações desse livro, pelo excelente motivo de que, quando discordamos no passado, no final acabei aceitando a opinião de Alistair Cockburn. Ele também mantém uma página na Web, no endereço **http://usecases.org**. [Constantine e Lockwood] fornecem um convincente processo para derivar interfaces com o usuário a partir de casos de uso; veja também o endereço **http://foruse.com**.