

Escrevendo User Stories Mais Efetivas

Nós mostramos como definir, de forma informal, user stories nas seguintes videoaulas: Dicas para Criar Informalmente User Stories, Receita para Criar Informalmente User Stories, Exemplo de Criação Informal de User Stories: Parte 1 e Exemplo de Criação Informal de User Stories: Parte 1. O papel dos 3 Cs, em especial da Conversação, é essencial para se produzir código de qualidade a partir das user stories desenvolvidas informalmente. Mas nem sempre isso acontece de forma fácil. Muitas vezes as user stories não são muito boas e tornam o trabalho do time de desenvolvimento mais difícil e menos produtivo.

Bill Wake, com objetivo de tornar o processo de definição mais formal e tentar garantir user stories mais efetivas, que facilitem o trabalho posterior de implementação, propôs o INVEST, um conjunto de diretrizes mais formais para definição de user stories.

Definição

INVEST é um acrônimo que ajuda a lembrar 6 critérios ou diretrizes amplamente aceitos. Forma, de fato, uma lista de verificação para avaliar a qualidade de uma história de usuário. Se a história não atender a um desses critérios, a equipe pode querer reformulá-la, ou mesmo considerar uma reescrita da história, que muitas vezes se traduz em rasgar fisicamente o velho cartão de história e escrever um novo.

Assim, uma boa história de usuário deve seguir as seguintes diretrizes:

- "I" – Independente de todas as outras user stories quanto ao valor de negócio (BV ou Business Value)
- "N" – Negociável: não é um contrato definitivo para o requisito; de fato, é apenas o ponto inicial!
- "V" – Valiosa para o cliente, PO ou stakeholder: O BV é dado pelo PO, em nome de clientes e stakeholders
- "E" – Estimável, com uma boa aproximação;
- "S" – (Small) Pequena, de modo a caber dentro de uma iteração
- "T" – Testável: em princípio, mesmo se ainda não houver um teste especificado para ela

Explicação e Exemplos

Suponha um app para tirar foto e identificar animal na natureza! O tipo do usuário é um usuário comum, que tem o seguinte fluxo de ações para essa necessidade:

- ☑ Tirar foto de animal
- ☑ Examinar lista de potenciais animais identificados: cabe ao usuário decidir qual animal identificado é mais próximo do fotografado, se algum da lista!

Temos 2 user stories associadas a esse fluxo:

US1: "Como usuário, quero tirar foto de animal"; tem vários motivos para tirar fotos de animal; por exemplo, para guardar ou para identificar, quando não for conhecido! Ou seja, esta história faz parte de dois fluxos de ações e valor de negócio próprio em cada caso.

US2: "Como usuário, quero examinar lista de potenciais animais identificados, de modo a tentar saber qual é o tipo do animal por conta própria". A US2 depende funcionalmente da US1 e depender funcionalmente não tem problema! O que tem problema é uma história ser dependente do **valor de negócio ou BV de outra história de usuário**. Por Independente, queremos mostrar que o valor de negócio de uma história é independente do valor de negócio de qualquer outra história de um dado aplicativo. Ou seja, sendo independentes, poderemos implementar cada user story de forma independente porque isso faz sentido para o PO.

Suponha, agora, que eu tenha outro fluxo para essa necessidade:

- ☑ Tirar foto de animal
- ☑ Examinar lista ordenada de potenciais animais identificados, do mais provável para o mais improvável: cabe ao usuário confirmar que o animal identificado como mais provável é realmente mais próximo do fotografado; se não, passar para o próximo da lista ordenada e assim por diante!

Temos 2 user stories associadas a esse fluxo:

US1: "Como usuário, quero tirar foto de animal"

US3: "Como usuário, quero examinar lista ordenada de potenciais animais identificados, do mais provável para o mais improvável, de modo a saber com um certo grau de probabilidade o seu tipo"

US3 é independente de US1 e também de US2; US2 e US3 consistem em ações bem próximas, mas distintas. Eventualmente, US2 poderia até ser a parte inicial de US3, que usa critérios de casamento de imagens para entregar uma lista ordenada, do mais provável para o mais improvável. Ou seja, cada uma tem o seu valor de negócio, embora US3 dependa funcionalmente de US2 nessa eventualidade levantada.

O PO pode acreditar que US2 tenha um valor de negócio mais alto que o do US3 e decide implementar essa história logo no primeiro sprint para o primeiro release do app, pois um usuário típico e alvo principal do app (usuário estudioso da área) pode achar importante ter uma lista de animais potenciais e ir usando seu conhecimento para confirmar o seu achado. O PO também pode acreditar que o usuário leigo, não estudioso, não vai se interessar pela US2 e sim pela US3, mas como ele prevê que esse tipo de usuário leigo vai ser minoria no uso do app, ele dá um BV baixo para a US3, que será implementado em algum sprint futuro para uma release futura do app!

Mas o time de desenvolvimento consegue convencer o PO que a US3 vai ser útil para todos os tipos de usuários e será mais prático para o usuário estudioso confirmar suas habilidades começando da proposta mais provável indicada pelo sistema. Isso constitui o uso do critério Negociável: time de desenvolvimento conversa e negocia com o PO o seu ponto de vista entre US2 e US3!

A diretriz Valiosa foi tratada nos exemplos de Independente e Negociável acima. Vou mostrar então um contraexemplo:

"Como usuário, eu quero testar o método de procura e casamento de imagens, de modo que eu saiba que ele funciona"

Vamos lembrar que uma user story representa o "quê" deve ser feito, do ponto de vista do PO/cliente/stakeholder, enquanto sua implementação será fruto de "como" isso deverá ocorrer mais à frente, quando o time de desenvolvimento quebrar a user story em tarefas (tasks) durante o sprint planning; ou seja, isso poderia ser tratado unicamente como uma tarefa técnica, que não tem um valor de negócio mensurável para o PO.

No entanto, o time de desenvolvimento pode acreditar que seria importante e vantajoso examinar a literatura e experimentar alguns algoritmos de reconhecimento e casamento de imagens antes de se fixar em algum específico. Cabe ao time tentar convencer o PO que, quanto melhor esse algoritmo funcionar, mais funcional e agradável ao usuário o app vai ficar.

Assim, a user story, neste caso chamada de técnica, pode entrar no Backlog de Produto, com valor de negócio bem alto e ser, por exemplo, implementada já no próximo sprint. Mas ela teria que ter uma cara de user story, identificando o tipo de usuário, que agora é técnico:

"Como time de desenvolvimento, queremos testar os métodos de procura e casamento de imagens, de modo que se saiba qual funciona melhor no contexto do app"

A cerca da diretriz Estimável, quando o time de desenvolvimento não consegue estimar direito o esforço ou tamanho de uma user story, temos geralmente dois motivos:

- ☑ A user story é tão grande em termos de tamanho/esforço de desenvolvimento que ela caracteriza um epic
- ☑ Não está claramente descrita ou embaralha muitas coisas, o que também a caracteriza como um epic

Quando apenas não está claramente descrita, uma saída é melhorar a sua descrição. Muitas vezes, por meio da conversação, a história fica mais clara e o time é capaz de estimar seu tamanho, que, se for razoável e pequeno em esforço, poderá ser implementável em um sprint de 2 semanas; outras vezes, fica mais claro que é um epic puro e ponto final.

Nos outros casos, resta quebrar as user stories em histórias menores em esforço de desenvolvimento; com isso, a descrição de cada uma fica mais clara e, em consequência, fica mais fácil estimar seu tamanho ou esforço de desenvolvimento.

Mas atenção: Nem sempre é conveniente quebrar um epic quando o BV dele não for tão prioritário. Pode-se deixar para fazer essa quebra quando o epic estiver para ser alvo da implementação. Neste caso, algumas das user stories derivadas do epic podem ser priorizadas para o próximo sprint, enquanto outras podem permanecer no Product Backlog para escolha futura!

A diretriz Small (Pequena) refere-se a user stories cujo esforço/tamanho de desenvolvimento implica em entregas dentro das 2 semanas de um sprint usual. Como corolário, quanto menor a US em esforço/tamanho de desenvolvimento, mais fácil para o time estimar seu tamanho, mais fácil para entender tudo que ela acarreta e mais fácil de quebrar em tarefas (tasks)!

Quanto à diretriz Testável, se a user story estiver sendo descrita com palavras que dificultam definir testes de aceitação (ATs – Acceptation Tests), deve-se reescrevê-las para possibilitar a definição de ATs correspondentes, mesmo que definir isso seja deixado para mais tarde no processo. Exemplo de palavras que dificultam definir ATs: rapidamente, simplesmente, facilmente etc.

Por exemplo, seja a seguinte história para o app:

“Como um usuário, eu quero ver a lista de felinos rapidamente, de modo que eu possa identificar o felino da minha foto”

O que significa “ver a lista rapidamente”? Como testar isso? E agora:

“Como um usuário, eu quero ver a lista de felinos em 15s, de modo que eu possa identificar o felino da minha foto”?

Agora, eu posso testar se a lista é apresentada em média dentro de 15s!!! Outro critério de aceitação pode ser o número de felinos da lista que são apresentados de cada vez, digamos 10 felinos por vez! Isso também pode ser testável.

Quando Usar o INVEST

O INVEST poderá ser usado nos seguintes eventos do Scrum&XP:

- ☒ Na Parte 2 do Sprint Zero, quando as primeiras user stories são levantadas e refinadas por meio do Workshop correspondente da Análise Ágil da Visão da Aplicação. Basta incluir o INVEST junto com as diretrizes informais propostas lá!
- ☒ Na Fase 1: Avaliação do PBL, do Sprint Planning, em que novas USs são eventualmente adicionadas pelo PO
- ☒ Durante um dado sprint, numa reunião chamada "Backlog Grooming" ou reunião de refinamento (grooming) do Product Backlog, usualmente com duração de 2 horas num sprint de 2 semanas, realizada na quarta ou quinta à tarde da segunda semana! O Backlog Grooming corresponde basicamente à realização do Workshop mostrado na Visão da Aplicação (Sprint Zero) junto com o emprego do INVEST, se for desejado!