

Product Backlog, Sprint Backlog e suas partes

Product e Sprint Backlogs

No capítulo passado comentamos que o P.O. deve chegar à reunião de Planning com o topo do Product Backlog devidamente atualizado, isto é, com os itens mais relevantes para o negócio, já granulares o bastante para caber no Sprint. Falamos também que sairemos de cada Planning Meeting com um Sprint Backlog. Faltou, no entanto, estudarmos mais a fundo o que é o Product e os Sprint Backlogs e quais regras do Scrum se aplicam a eles.

Product Backlog

Assim que imaginamos um projeto, já temos uma lista de funcionalidades que gostaríamos de ter nele! Essa lista é frequentemente usada para decidir se o projeto realmente será desenvolvido e, quando vamos começar a transformar esses desejos em realidade é importante fazê-lo de forma consciente, para maximizar o valor produzido a cada interação.

O Product Backlog, a lista ordenada de itens a serem feitos no projeto, é o artefato que ajuda o P.O. a manter todos esses pedidos organizados. O P.O. é responsável por mantê-lo atualizado e priorizado de modo a agregarmos o máximo de valor possível para o cliente a cada iteração.

Embora clientes, usuários e o próprio time tenham a liberdade de influenciar mudanças no Product Backlog, a única pessoa que de fato o altera é o Product Owner -- inclusive, o papel é chamado dessa forma porque o P.O. é dono do Product Backlog! Os itens do Product Backlog são sempre mutáveis: o P.O. pode adicionar itens, removê-los e re-priorizá-los sempre que achar que consegue agregar mais valor ao projeto dessa forma.

A proposta do Product Backlog é, também, que evitemos colocar trabalho em itens pouco importantes ou que sequer sabemos se serão feitos: os itens mais prioritários de um backlog têm que ter sua importância de negócio estudada e ser pequenos o bastante para que o time estime e os aloque em Sprints, mas os mais distantes e menos prioritários podem permanecer em blocos maiores e mais grosseiros, que serão refinados conforme sua prioridade cresce.

Sprint Backlog

Uma vez que itens cheguem ao topo do Product Backlog, eles serão discutidos em uma Planning Meeting, quebrados em alguns sub-itens técnicos e, se escolhidos para serem executados, esses itens e sub-itens comporão o chamado Sprint Backlog: a lista ordenada de itens funcionais e seus sub-itens técnicos que serão feitos durante essa Sprint.

Diferente do Product Backlog, apenas o time pode influenciar e alterar o Sprint Backlog. Existe uma razão para isso: o Sprint Backlog é formado dos itens mais prioritários do Product Backlog e é uma indicação séria de problemas sistêmicos se os itens mais prioritários precisarem sofrer grandes modificações. Se os itens mais prioritários não fizerem sentido nas, digamos, 2 semanas de Sprint, isso indica que o P.O. não fez um bom trabalho de refinamento e de descobrir com usuários o que eles realmente precisam.

Histórias e tarefas

Cada item que compõe um Product Backlog representa uma funcionalidade, algo que agrega valor para o usuário final -- note, portanto, que "documentação técnica" não é um item válido, já que o usuário não se beneficia disso.

Esses itens podem ter o formato que você quiser -- por exemplo, um conjuntinho de casos de uso do sistema pode ser um item válido. Há uma forte preferência entre agilistas, no entanto, de usar um formato especial para representar esses itens: uma história de usuário (*user story*).

Uma história é um formato criado em eXtreme Programming (XP) para representar um item que agrega valor a usuários e agrega, de uma forma bastante simples, três informações importantíssimas para a priorização e posterior desenvolvimento da funcionalidade: por que é importante, para quem é importante e, só então, o que a pessoa quer, em si. O modelo que costumamos preencher é o seguinte:

[TÍTULO]

```
**Para...** [por que o pedido é importante]
**No papel de...** [para quem é importante]
**Quero...** [o pedido em si]
```

Em um sistema de vagas online como o OndeTrabalhar.com, por exemplo, poderíamos ter a história:

VAGAS POR LOCALIDADE

```
**Para...** não perder tempo olhando cada vaga para descobrir se é na minha cidade
**No papel de...** pessoa procurando trabalho
**Quero...** ter a opção de filtrar as vagas de trabalho por cidade
```

Durante o Planning é comum verificarmos o entendimento da história quebrando ela em itens menores, técnicos, que não necessariamente agregam valor ao usuário individualmente. Esses sub-itens técnicos de histórias são chamados **tarefas**. A história acima poderia ser quebrada nas seguintes tarefas:

- Cadastro de cidades no banco de dados;
- Mudança no formulário de postagem de vaga para limitar à lista de cidades;
- Filtro de busca de vagas no menu principal.

Muito comumente também, embora não seja regra, é vermos histórias em cartões (fichas pautadas) e tarefas em post-its.