

A abordagem

# S.P.I.D.R

para quebra de  
histórias de usuário

## SPIKES

Faça de uma história grande algo menor executando um Spike, que é uma atividade de pesquisa onde o time ganhará mais conhecimento sobre a história.

Algumas vezes, apenas fazer um Spike faz com que o trabalho remanescente seja gerenciável. Em outras vezes, o novo conhecimento criado pelo spike faz com que seja mais fácil visualizar formas de quebrar a história.

## REGRAS

Algumas vezes, uma história fica grande por causa de regras de negócio, padrões de tecnologia ou coisas do tipo que são essenciais para a funcionalidade.

Considere flexibilizar a contemplação destas regras em uma versão inicial, e adicione-as em versões posteriores.



## DADOS

Procure por formas de quebrar a história baseado no tipo de dado que a funcionalidade suportará.

Pode uma primeira versão conter somente dados válidos e uma versão posterior dar suporte para dados inválidos? Que tal tipos de dados frequentes e dados menos frequentes?

## PATHS/ FLUXOS

Considere os fluxos dentro da história e quebre cada fluxo em uma história diferente.

Desenhe um fluxograma simples do que acontece na história. Cada sequência de passos pode ser uma história separada.

Pegue um passo grande de um fluxo e transforme numa história.

## INTERFACES

Quebre uma história em múltiplas interfaces se o suporte a estas interfaces faz com que a história tome um tempo significativamente maior para desenvolver.

Quebre histórias por browser ou por versão de SO, ou por hardware diferente.

Considere construir uma interface mínima ao usuário ou deixar a estilização de uma tela fora do escopo inicial.



MOUNTAIN GOAT  
SOFTWARE

Original em inglês: betteruserstories.com

Tradução por Priscila Chagas priscilachagas.com