



Boas Histórias de Usuários

Marcelo Werneck

Começar com objetivos

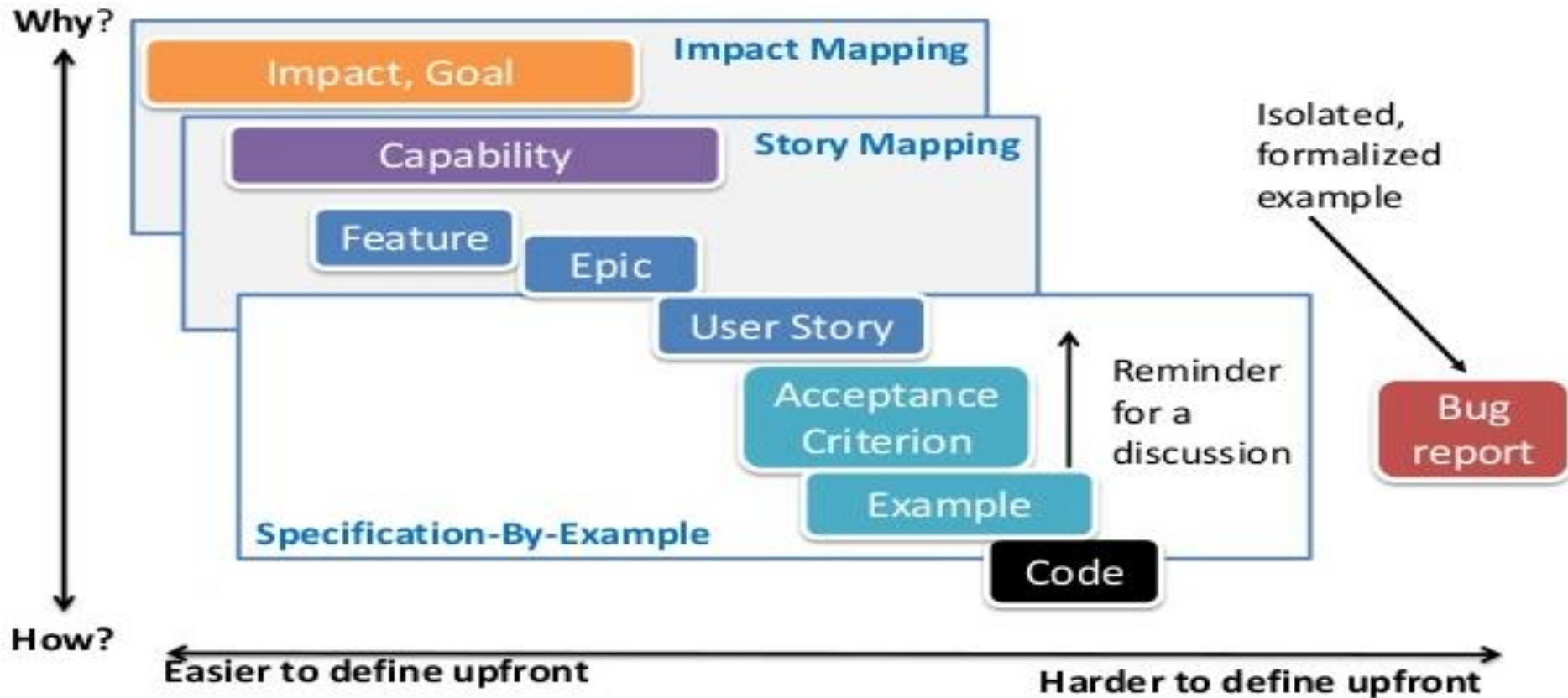
Em projetos grandes com muitos usuários

Difícil saber por onde começar

Sugestão é considerar cada papel e começar pelos seus objetivos de maior prioridade.

Objetivos são histórias de alto-nível.

Irão gerar histórias adicionais.



Fatiar o bolo

Há várias maneiras de se quebrar uma estória em pedaços menores. Existe tendência de se quebrar considerando aspectos técnicos.

Melhor abordagem: Fatiar o bolo.

Cada estória deve conter algum nível completo de funcionalidade. Uma maneira também de se testar camadas da arquitetura.

Fatiar o bolo

Aplicação pode até ser entregue com funcionalidade parcial mas que inclui todas as camadas.



<http://www.noticiasaoiminuto.com/lifestyle/500888/esta-e-a-forma-mais-segura-de-cortar-um-bolo-redondo>

PUC Minas **Virtual**

Ciclos trimestrais

Não há regra que determine quanto o projeto deve durar.

Para projetos mais longos que quatro meses, uma boa sugestão é usar ciclos trimestrais

Lançar uma versão testada e documentada a cada três meses.

Restrições em cartões

Devem ser obedecidas e não implementadas diretamente.

Não são alocadas a iterações e não são diretamente estimadas.

Podem ser colados na parede.

Podem ter testes de aceitação associados.

Algumas coisas não são histórias

Formato de histórias de usuário não é apropriado para tudo.

Se for necessário documentar requisitos em outro formato, então use formato adequado.

Ex. documentos de interface

Documentos de fornecedores externos também podem demandar outros formatos.

Escreva na voz ativa

Estórias de usuário são mais fáceis de ler e entender quando escritas na voz ativa.

Em vez de:

- Um currículo pode ser publicado por um recrutador.

Escreva:

- Um recrutador pode publicar um currículo.

Não numere os cartões

Pode ser uma tentação para manter controle dos cartões ou ter algum mecanismo de rastreabilidade.

Acrescenta esforço desnecessário.

Normalmente histórias são referenciadas pela funcionalidade em vez de por um número.

Se sentir a necessidade, tente acrescentar um título para a história.

Problemas



Histórias pequenas demais

Histórias muito pequenas causam problemas de estimativas quando a ordem da implementação afeta consideravelmente as estimativas.

- Resultados da pesquisa devem ser salvos em um arquivo XML.
- Resultados da pesquisa devem ser salvos em um arquivo HTML.

Histórias pequenas demais

Ao se colocar a história em uma iteração, ela pode ser quebrada, mas antes disso, as histórias deveriam ser combinadas (para efeitos de planejamento).

Histórias dependentes

Pode haver dificuldade no planejamento das iterações quando há histórias dependentes. A inclusão de uma história em uma iteração acarreta a inclusão de outra relacionada. Pode acontecer quando as histórias são muito pequenas ou foram divididas inapropriadamente. Solução pode ser combinar histórias. Lembrar da divisão em camadas (fatias do bolo)

Goldplating

Acontece quando são entregues funcionalidades que não foram solicitadas ou além do que é necessário para sua implementação.

Difícil impressionar o cliente com a participação constante em projetos ágeis.

Pode-se evitar aumentando a visibilidade das tarefas de cada um e reuniões de término de iteração.

Detalhes demais

Acontece quando muito tempo é gasto com detalhes antes da implementação da história ou ainda com a escrita da história.

Demonstra algo valor para documentação em detrimento da conversação.

Uso de cartões (espaço limitado)

Iniciar desenvolvimento sem Visão

É um erro iniciar o desenvolvimento sem nenhuma visão do produto.

Deve haver uma visão disponível para enunciar claramente o cliente selecionado, suas necessidades e atributos críticos

Esta visão ajuda a determinar quais recursos deverão ser implementados e garante a criação de um produto útil.

Especificação de Requisitos disfarçada

Uma especificação disfarçada em Product Backlog é tentadora porque atende nosso desejo de saber todos os requisitos de cara
Mas não tem suporte para surgimento de novos requisitos.

Não vê requisitos com algo fluido e transitório.
Pode ser sinal de relacionamento pouco saudável entre Product Owner e equipe.

Contratos de Preço fixo

Se for possível, evite projetos com preço e escopo fixos.

Se não for possível, divida um contrato de preço fixo em duas partes.

O primeiro cria a visão do produto e implementa-o parcialmente.

O segundo continua a dar vida ao projeto com base em feedback do cliente.

Referências bibliográficas

COHN, M. User stories applied, Addison-Wesley.