

SCRUMBUT

Aula de Luiz Eduardo Guarino de Vasconcelos

Agenda

- ScrumBut
- Mitos ágeis



Você utiliza Scrum ou não?

Introdução

- A resposta é sim?
 - ▣ Então segue todas as práticas, papéis, artefatos, regras.
- A resposta é não?
 - ▣ Provavelmente você está adaptando o Scrum
 - ▣ ScrumBut ou Scrum And.

Introdução

- ScrumBut significa que o processo Scrum possui disfunção que está contribuindo para o problema, e é muito difícil de corrigir.
 - ▣ Equipe tenta adaptar o Scrum
- Problema passa a ser desconsiderado ou torna-se invisível.
- Principal consequência é a perda total dos princípios do Scrum e das metodologias ágeis.

Sintaxe do ScrumBut

- (ScrumBut) (Razão) (Solução)
- (prática do Scrum não aplicada) (desculpa ou motivo) (solução alternativa para deixar invisível a disfunção Scrum)

Exemplos

- Sua empresa usa SCRUM?
- R: Sim, mas a Daily Scrum todos os dias sobrecarrega a equipe e não é produtiva, de modo que só temos uma por semana.
- R: Sim, mas não fazemos as retrospectivas pois já temos um processo maduro na equipe de desenvolvimento.
- R: Sim, mas nossas sprints variam de 6 a 8 semanas.

Porque adotam o ScrumBut?

- ❑ Clientes não querem ser envolvidos, simplesmente querem o produto pronto
- ❑ A empresa não está disposta a mudar esses processos
- ❑ A empresa não quer ser envolvida e não respeita as prioridades do PO.
- ❑ Não foi possível convencer o gerente de TI dos benefícios de permitir que a equipe Scrum ser autogerenciável
- ❑ Nem todos da equipe estão disponíveis para trabalharem de forma integral para o projeto ou Sprint.
- ❑ A equipe não sente necessidade das reuniões diárias
- ❑ Nem todos da equipe tem condições de se responsabilizar por funcionalidades

Porque adotam o ScrumBut?

- ❑ Nem sempre conseguem reunir todos da equipe nas retrospectivas
- ❑ O Scrum Master não respeita as decisões da equipe e decide o que cada um deve fazer
- ❑ O Scrum Master não está comprometido com Scrum e não remove os obstáculos enfrentados pela equipe
- ❑ O PO não consegue priorizar as atividades

Riscos e consequências

- A equipe quando não adota todas as práticas Scrum, passa a assumir riscos e, muitas vezes, com graves consequências:
 - ▣ Perda de tempo
 - ▣ Conflitos internos
 - ▣ Clientes insatisfeitos
 - ▣ Falta de gestão
 - ▣ Perda do controle
 - ▣ Atraso na entrega

Riscos e consequências

- Diversas empresas modificaram o Scrum e afirmam ter conseguido ter benefícios.
 - ▣ Faça com cautela.
 - ▣ Complemente com XP, Kanban, Lean.
- Mudanças são inevitáveis e fundamentais para o sucesso do time.

Como perceber o ScrumBut?

- Time sem objetivos
- Time sem foco, trabalhando sob demanda
- Trabalho constante e sem colaboração do time
- Destaques ou heroísmos individuais
- Falta de gestão de risco
- Falta de compromisso com melhoria contínua nas retrospectivas
- Equipes desconectadas
- Gestão individual ou gestão por cronograma e atividades

Como perceber o ScrumBut?

- ❑ Atrasos constantes
- ❑ Entregas com mais de 4 semanas
- ❑ Reuniões canceladas constantemente
- ❑ Daily scrum cansativas e sem produtividade
- ❑ Daily Scrum duram mais de 15 minutos
- ❑ SM desconectado do Time
- ❑ Constantes interrupções das sprints
- ❑ PO não respeita estimativas do Time

Como evitar ScrumBut?

- ❑ Discussões regulares sobre os valores do Scrum
- ❑ Discussões regulares sobre os princípios do manifesto ágil
- ❑ Debates sobre o que o Scrum significa para cada membro
- ❑ Inspeções constantes sobre os benefícios e resultados obtidos com Scrum
- ❑ Permitir que o Time se autogerencie dentro de um time-box
- ❑ Questione-se regularmente se os processos Scrum estão sendo respeitados
- ❑ Identifique e evite qualquer ação de heroísmo individual

Mitos ágeis

- Ágil não documenta
 - Você acredita que um software “sério” pode ser documentado com aqueles papezinhos do Kanban?
 - Documentação é diferente de Especificação!
 - Especificação é tudo aquilo que é necessário para que informação suficiente seja transmitida de um emissor (que precisa de um determinado software ou funcionalidade) a um receptor (que vai desenvolver o software ou funcionalidade)
 - Pode ser um documento longo e detalhado
 - Pode ser uma observação in loco do programador
 - Pode ser a explicação de um solicitante ao programador
 - O que é necessário, é desenvolver o software corretamente!

Mitos ágeis

- Ágil não documenta
 - ▣ Documentação é tudo aquilo necessário, e suficiente, para que alguém entenda como um sistema de software funciona.
 - Existem diferentes formas de documentar e cada uma serve para um propósito específico.
 - Exemplos: UML, Manuais, Materiais de treinamento, fotografias, código-fonte.
 - ▣ Idealmente, a documentação deve refletir, a qualquer momento, o estado atual de um software.
 - Quais documentos representam o software no estado atual?
 - Qualquer documentação que não represente isso, não serve para muita coisa.
 - Qual o esforço para manter a documentação atualizada?
 - Até quando estará atualizada?

Mitos ágeis

- Ágil não documenta
 - É muito comum a ideia de que normas como a lei Sarbanes-Oxley, Basel III ou FDA CFR 21 part 11 nunca são atingidas através de processos ou práticas ágeis
 - Não há nada no Scrum Guide que limite ou restrinja a produção de documentação
- “Software funcionando sobre documentação extensiva” ... “mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda”.
- Devemos entregar o software funcionando e o que mais for necessário (de acordo com cada cliente)

Mitos ágeis

- Ágil só serve para tornar a vida do desenvolvedor mais fácil
 - ▣ Muitos dos conceitos ágeis são contra intuitivos
 - ▣ O conceito de auto-organização pode parecer um passo na direção da anarquia.
 - ▣ Auto-organização não é falta de organização
 - Um gerente consegue garantir que tudo será feito no prazo e no custo combinado?

Mitos ágeis

- Ágil só serve para tornar a vida do desenvolvedor mais fácil
 - Se não houver alguém responsável pelo projeto cuja função é cobrar datas e entregas, o projeto não terá sucesso?
 - Os pares sabem o que tem que ser feito
 - Mais difícil inventar desculpas nas reuniões diárias, onde um está frente ao outro
 - Todos tem responsabilidade sobre as funcionalidades

Mitos ágeis

- Na minha empresa isso não vai funcionar, aqui as pessoas precisam ser controladas
 - ▣ Quando precisamos de um médico, procuramos um bom médico. Quando levamos o nosso carro a um mecânico, escolhemos aquele indicado por algum amigo como sendo capaz de resolver qualquer problema. Quando vamos desenvolver software, isso muda. Achamos que um profissional pouco qualificado pode fazer um bom trabalho, contanto que siga o processo à risca.
- A equipe ser responsável pelo seu próprio trabalho traz benefícios que vão além dos ganhos no projeto, como diminuição do **turn-over** (demissões e contratações) e do **burn-out** (esgotamento profissional) e **melhoria da Percepção de Suporte Organizacional** pelos envolvidos (PSO)