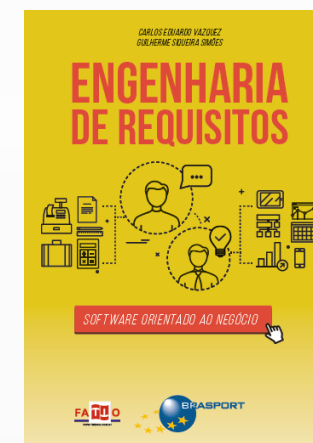


# Testes de software em processos ágeis



apoiar nossos clientes no  
**planejamento e avaliação**  
**de desempenho de**  
**processos de TI para**  
**alavancar o sucesso do seu**  
**negócio**



- ☐ Objetivos dos testes
- ☐ Agilidade e SCRUM
- ☐ Papel do testador
- ☐ Testes no contexto SCRUM
- ☐ Automação e integração contínua
- ☐ Independência de testes

# Objetivos do teste

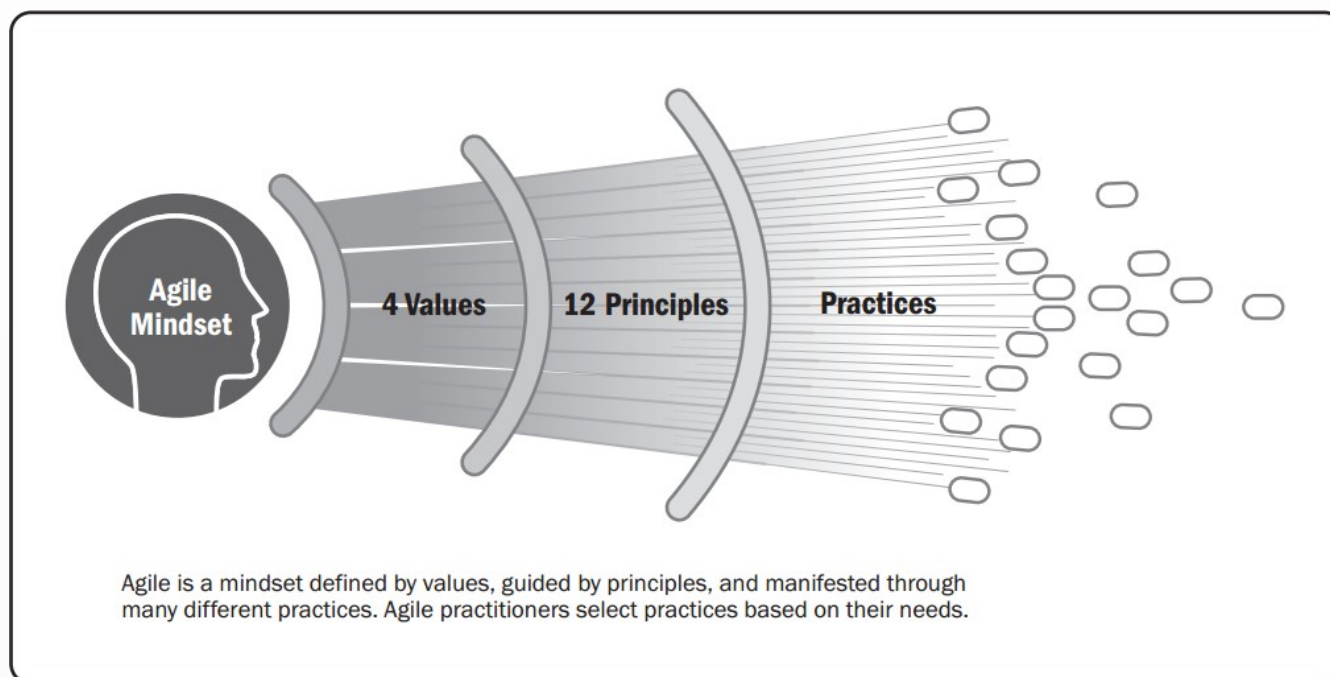
- ☐ Evitar que defeitos cheguem ao usuário
- ☐ Verificar se todos os requisitos especificados foram cumpridos
- ☐ Verificar se o produto está completo e validar se funciona como os interessados esperam
- ☐ Criar confiança no nível de qualidade do produto
- ☐ Verificar conformidade do produto a normas/leis/regulamentos
- ☐ Fornecer informações aos interessados para que tomem decisões a respeito da qualidade do produto

# O Manifesto Ágil (2001)

<http://agilemanifesto.org/iso/ptbr/manifesto.html>



- ❑ **Indivíduos e interações** mais que processos e ferramentas
- ❑ **Software em funcionamento** mais que documentação abrangente
- ❑ **Colaboração com o cliente** mais que negociação de contratos
- ❑ **Responder a mudanças** mais que seguir um plano

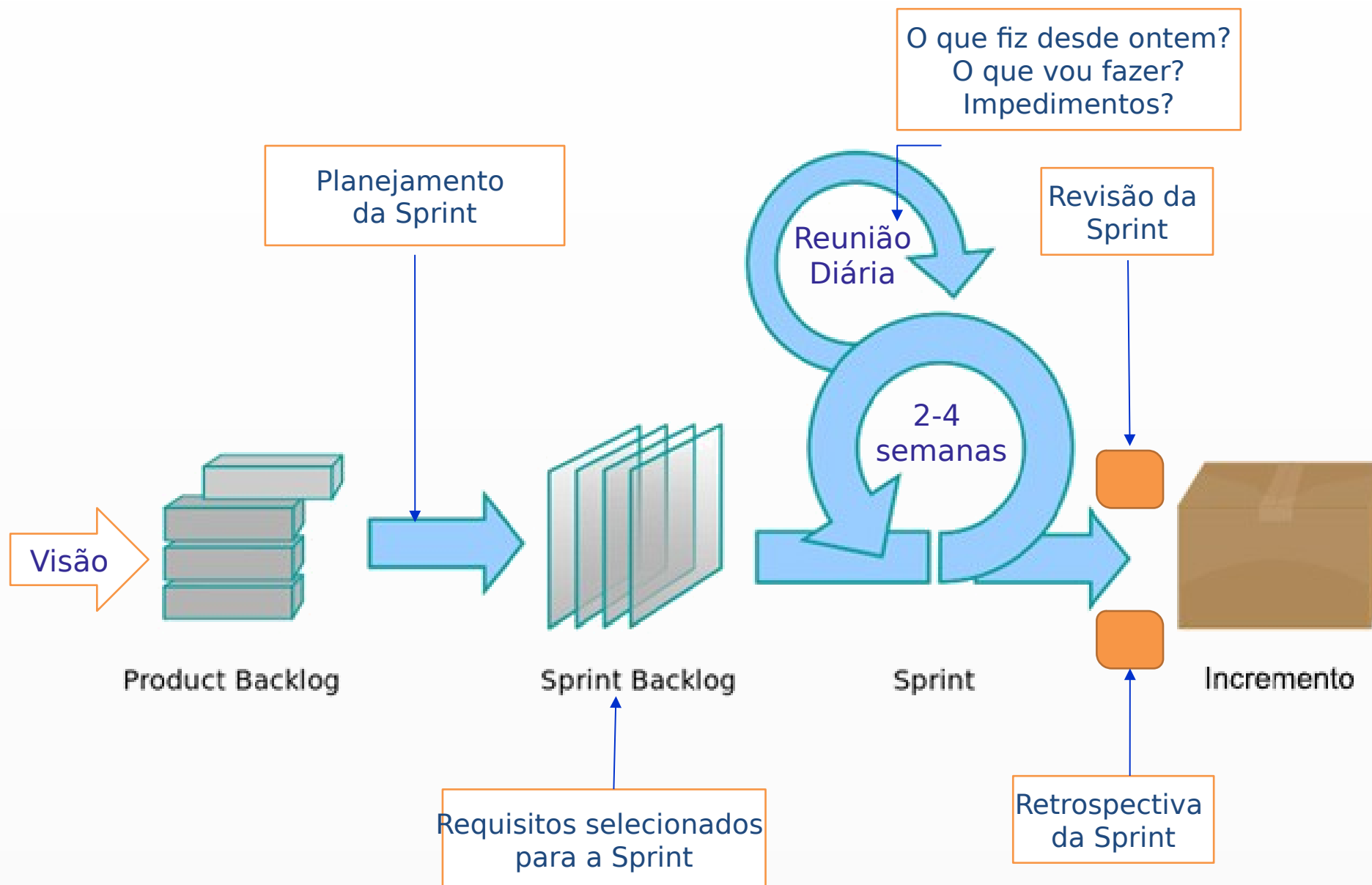


Há uma variedade grande de abordagens ágeis para o desenvolvimento de software

Na prática é comum a combinação de mais de uma abordagem

- ❑ A cada iteração concluída, o produto cresce. Por conseguinte, os testes também aumentam. E não basta testar as alterações no código na iteração atual, é preciso verificar se alguma regressão foi introduzida nas funcionalidades entregues em iterações anteriores
- ❑ Responder à mudança é um princípio chave ágil, as mudanças também podem ser feitas em funcionalidades anteriormente desenvolvidas. Logo, o risco de regressão no desenvolvimento ágil é alto
- ❑ É fundamental que se invista em automação de testes em todos os níveis o mais cedo possível. E que todos os ativos de teste, tais como testes automatizados, casos de teste manuais, dados de teste e outros artefatos de teste sejam mantidos atualizados com cada iteração

# SCRUM: Visão Geral



# Papéis – Time SCRUM



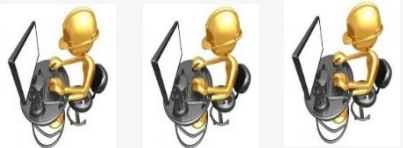
1. Product Owner, Equipe de Desenvolvimento e Scrum Master



3. Equipes auto organizáveis escolhem a melhor forma de completar seu trabalho, em vez de serem dirigidos por outros de fora da equipe



5. Equipes pequenas, de até 9 pessoas



7. A qualidade é responsabilidade de todo time

2. Os times são auto organizáveis e multifuncionais

4. Equipes multifuncionais possuem todas as competências necessárias para completar o trabalho sem depender de outros que não fazem parte do time

6. Entregam produtos de forma iterativa e incremental, maximizando as oportunidades de feedback

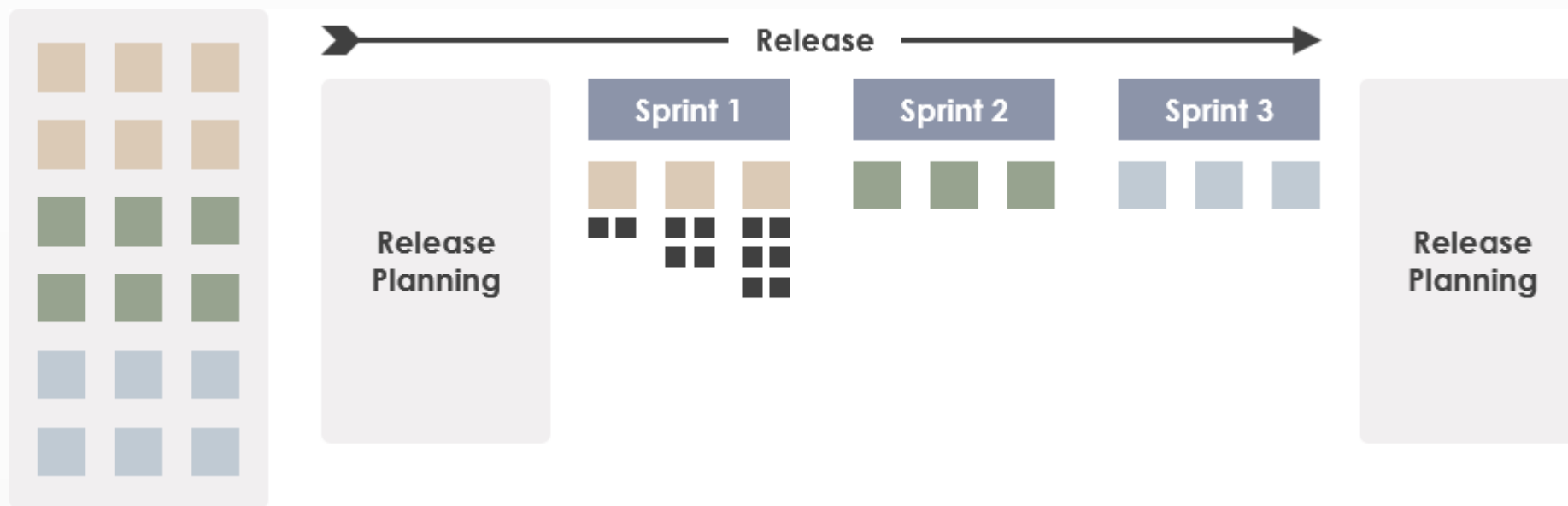


- ❑ Em uma abordagem convencional, cada pessoa tem um papel específico: analista de requisitos, arquiteto, programador, testador, documentador, etc
  - O processo induz a que cada um olhe apenas para seu quadrado
  - Sendo comum o jogo de empurra “isso é problema da especificação/arquitetura/construção”
  
- ❑ Num time ágil evita-se rotular de forma diferente os seus integrantes. Todos devem estar igualmente comprometidos com a entrega. A falha de um é a falha de todos. Não existe sucesso de um, apenas sucesso de todos
  
- ❑ Isso minimiza uma situação comum no desenvolvimento tradicional: o cabo de guerra entre o time de desenvolvimento e o time de testes

- ❑ Melhorar a história do usuário, identificando detalhes ausentes ou requisitos não-funcionais, formulando perguntas abertas aos representantes de negócio sobre a história do usuário, propondo formas de testá-la, e confirmando os critérios de aceite
- ❑ Colaborar com os representantes de negócio para ajudá-los a criar testes de aceite adequados
- ❑ Colaborar com desenvolvedores para chegar a acordos sobre a estratégia de teste, e decidir sobre as abordagens de automação de testes
- ❑ Transferir e ampliar o conhecimento de testes para outros membros da equipe e influenciar o desenvolvimento do produto
- ❑ *Pair programming / Pair testing* (XP): duplas de desenvolvedor+testador ou dupla de testadores
  - Testadores podem estar envolvidos na depuração de código e teste de componentes

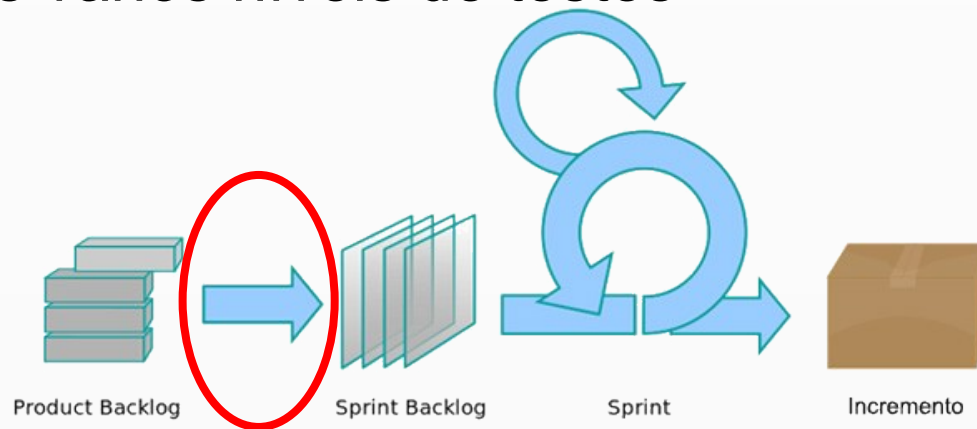
# Planejamento da Release (meses)

- ❑ Definição das histórias de usuários testáveis, incluindo os critérios de aceite
- ❑ Participação no projeto e análise do risco da qualidade
- ❑ Estimativa de esforço de teste associado às histórias do usuário
- ❑ Definição dos níveis necessários de teste
- ❑ Planejamento do teste para a release



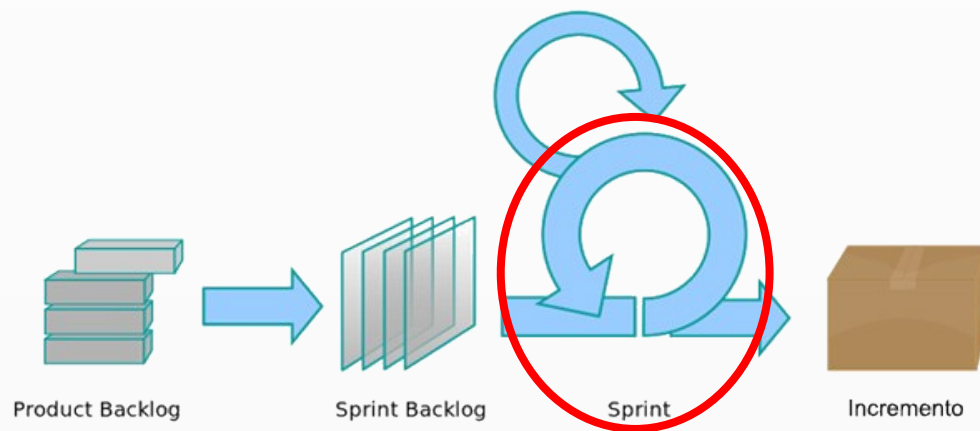
# Planejamento da Sprint (semanas)

- ☐ Avaliar a testabilidade das histórias de usuários
- ☐ Criar testes de aceite para histórias de usuários
- ☐ Análise detalhada de risco das histórias de usuários
  - Esta análise pode influenciar a sequência de desenvolvimento, assim como a prioridade e profundidade do teste das funcionalidades
- ☐ Dividir as histórias de usuários em tarefas (particularmente tarefas de teste)
- ☐ Estimar o esforço de teste para todas as tarefas de teste
- ☐ Identificar os aspectos funcionais e não funcionais do sistema a ser testado
- ☐ Apoiar e participar na automação de testes nos vários níveis de testes



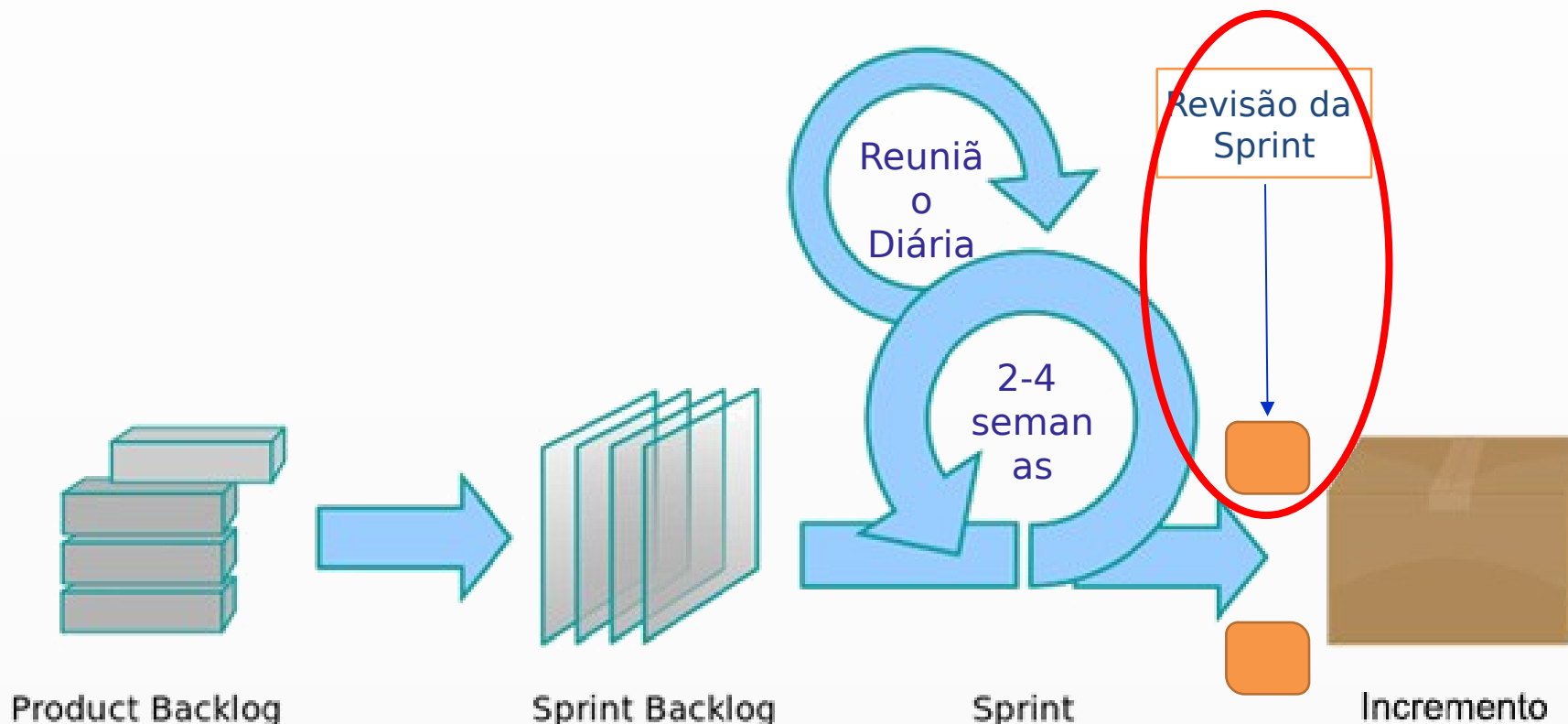
# Desenvolvimento da Sprint

- ❑ Teste de unidade, normalmente feito pelo desenvolvedor
- ❑ Testes de verificação de funcionalidades, que muitas vezes é automatizado, podem ser feitos por desenvolvedores ou testadores, e envolve testes contra os critérios de aceite da história do usuário
- ❑ Rever casos de teste manuais e automatizados de iterações anteriores e atuais para selecionar casos de teste que podem ser candidatos ao teste de regressão, e para retirar casos de teste que não são mais relevantes
- ❑ Testes de regressão: se reexecutam os testes de unidade automatizados e testes de verificação da funcionalidade da iteração atual e iterações anteriores, através da integração contínua



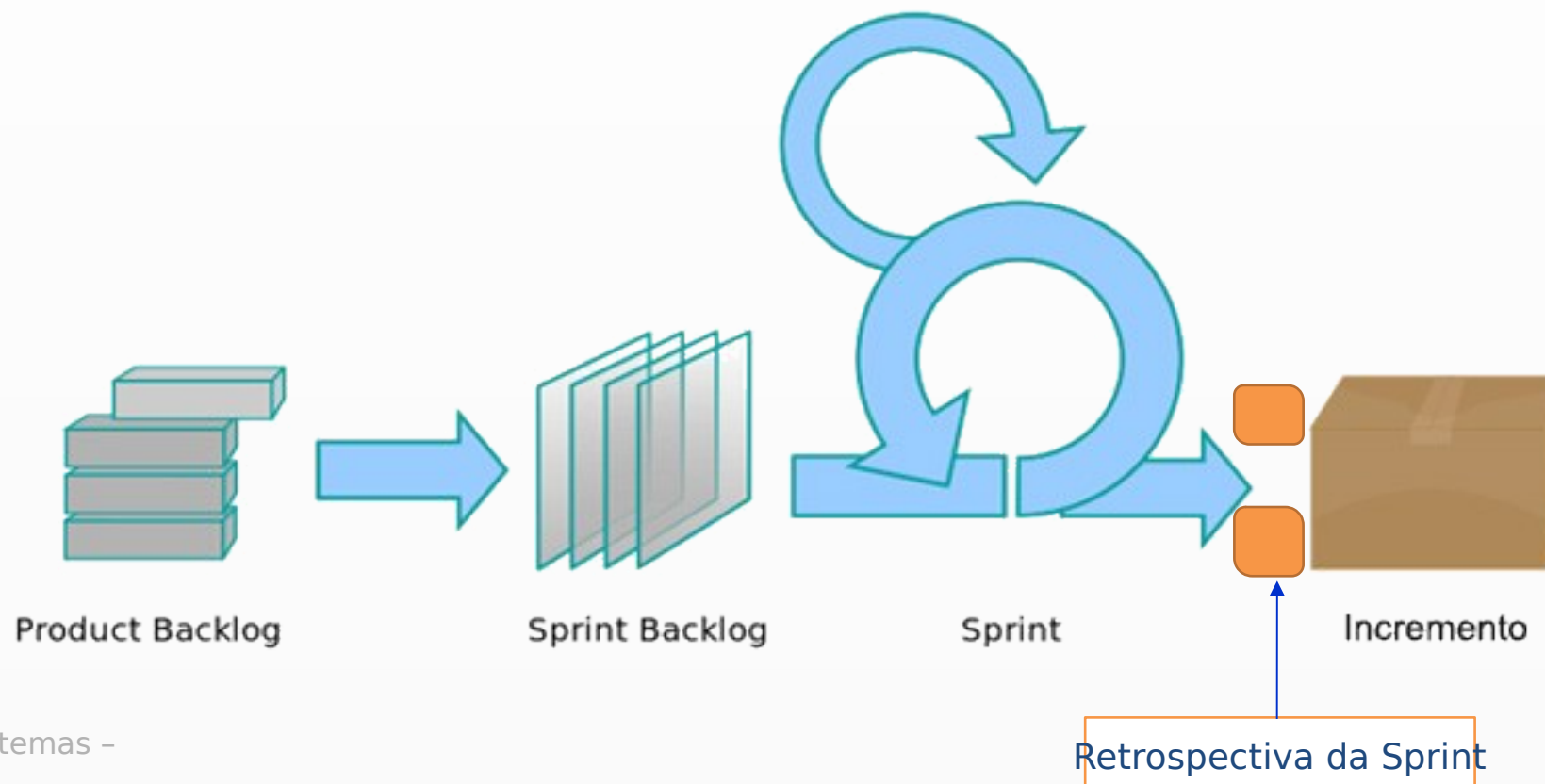
# Revisão da Sprint

- ❑ Testes de validação de funcionalidades, que normalmente é manual e pode envolver desenvolvedores, testadores e stakeholders que trabalham de forma colaborativa para determinar se a funcionalidade está apta para uso, para ter visibilidade do progresso realizado e receber feedback real dos stakeholders



# Retrospectiva da Sprint

- As retrospectivas podem resultar em decisões de melhoria relacionadas com o teste, focadas na eficiência, produtividade, qualidade do caso de teste e satisfação da equipe. Eles também podem abordar a capacidade de teste dos aplicativos, histórias de usuários, funcionalidades ou interfaces do sistema. Análise de causa raiz de defeitos podem conduzir testes e desenvolvimento de melhorias



- ❑ O conceito de pirâmide teste é baseado no princípio de testes de controle de qualidade (ou seja, a eliminação de defeitos o mais cedo possível no ciclo de vida)
- ❑ A pirâmide de teste enfatiza muitos testes para os níveis mais baixos (base da pirâmide) e, conforme o desenvolvimento se move para níveis superiores, o número de testes diminui (topo da pirâmide): testes de unidade, integração, sistema e aceite
- ❑ Além dos testes, as seguintes tarefas também podem ser automatizadas:
  - Geração de dados de teste
  - Carregamento dos dados de teste nos sistemas
  - Implantação de compilações para os ambientes de teste
  - Restauração de um ambiente de teste (p. ex., os arquivos de dados do banco de dados ou site) para uma linha de base
  - Comparação das saídas de dados



- ❑ A gestão da configuração, compilação, construção de software, implantação e testes são envolvidos em processo único, repetitivo e automatizado. Como os desenvolvedores integram seu trabalho constantemente, constroem constantemente e testam constantemente, os defeitos no código são detectados mais rapidamente
- ❑ Análise estática de código: execução de análise estática de código e relatórios de resultados
- ❑ Teste da unidade: executar os testes de unidade, verificando a cobertura de código e relatando os resultados dos testes
- ❑ Implantação: instalar o projeto em um ambiente de teste
- ❑ Teste de integração: executar os testes de integração e relatar os resultados.
- ❑ Um processo automatizado de construção e teste ocorre diariamente e detecta erros de integração de forma rápida e antecipada. A integração contínua permite aos testadores executarem testes automatizados regularmente, em alguns casos como parte do próprio processo de integração contínua, e enviar rapidamente o feedback para a equipe sobre a qualidade do código. Estes resultados dos testes são visíveis para todos os membros da equipe, especialmente quando os relatórios automatizados são integrados ao processo

# Organização dos Testes



- ❑ Times ágeis são compostos muitas vezes por um ou mais testadores
  
- ❑ Outra abordagem consiste em ter uma equipe de teste separada e independente da equipe de desenvolvimento, podendo
  - Alocar testadores sob demanda em momentos pontuais da sprint
  - Alocar testadores no início do projeto, numa estratégia de mais longo prazo
  - Alocar testadores especializados para atividades independentes da sprint, tais como o desenvolvimento de ferramentas de teste automatizado, a realização de teste não funcional, criar e suportar ambientes de teste e de dados

# Independência dos Testes



- ❑ Um princípio chave de testes é a independência de testes
- ❑ Deve haver certa segregação sobre algumas atividades de testes, de forma que quem desenvolve não seja inteiramente responsável por também testar
- ❑ Riscos:
  - Os testadores trabalham tão estreitamente com os desenvolvedores que eles perdem a mentalidade apropriada de um testador (viés de confirmação)
  - Os testadores se tornam tolerantes ou omissos sobre práticas ineficientes, ineficazes, ou de baixa qualidade na equipe
  - Conflito de interesse: realizar a entrega da sprint ou apontar problema?

# Conclusão



- ❑ Os objetivos do teste continuam os mesmos seja desenvolvimento ágil ou tradicional
- ❑ Especificações mais enxutas de requisitos implicam envolvimento do profissional de testes desde o início do desenvolvimento
- ❑ Intensidade de mudanças gera risco maiores de regressão a cada iteração
  - Mitigação via automação de testes em todos os níveis
  - E integração contínua
- ❑ A maior parte do tempo do profissional de testes está dedicada à criação, monitoramento, manutenção de testes automatizados
  - Necessidade de sólida formação técnica em automação
- ❑ A integração do profissional de testes com o time de desenvolvimento pode por em risco a independência dos testes
  - Mitigação via diferentes formas de organização dos testes



## COMO PODEMOS TE AJUDAR



- Serviços
  - Fábrica de Testes (Outsourcing)
  - Automatização do processo de inspeção e testes
  - Mentoria em testes para a equipe do cliente



# PRÓXIMOS EVENTOS



- Não culpe o termômetro pela febre: benefícios colaterais da medição de software
  - Data: 26/01/2021 às 13 horas (Horário de Brasília)
  - Inscrições gratuitas em: <https://bit.ly/3p3YNYn>





# Apresentador

GUILHERME SIQUEIRA SIMÕES

[guilherme.simoes@fattocs.com](mailto:guilherme.simoes@fattocs.com)

Linkedin:

[br.linkedin.com/in/guilhermesimoess](https://br.linkedin.com/in/guilhermesimoess)

Skype: [guilherme.s.simoes](https://www.skype.com/people/guilherme.s.simoes)

Whatsapp: +5527981117505

