

Modelos, Métodos e Técnicas de Engenharia de Software

Prof.: Sônia A. Santana



Projeto

Projeto é um esforço temporário empreendido para criar um produto, serviço ou resultado único (PMBOK).

Temporário

- Início
- Término

Entregas exclusivas

- Produto
- Serviço
- Resultado

Exemplos de Projetos

- Novo produto
- Nova filial
- Novo negócio
- Nova tecnologia

Gerenciamento de Projetos

- Gestão de projeto é o **processo de tomar decisões** que envolvem a utilização de recursos para realizar atividades temporárias, com o objetivo de atingir um resultado específico.
- Ao longo dos últimos 30 anos o PMI, através do guia PMBOK®, se tornou referência mundial em gestão de projetos.

Áreas de Conhecimento PMBoK



Os cinco grupos de processos do PMBOK são: Iniciação, Planejamento, Execução, Monitoramento e Controle e Encerramento.

Gerenciamento de Projetos

- Orientado a processos
 - Processos bem definidos devem ser impostos para garantir a qualidade do produto
- Rígido
 - Pressupõe que é possível especificar de antemão todos os requisitos do projeto
- Preditivo
 - Cada etapa de desenvolvimento é baseada na etapa anterior, parte do princípio de que requisitos são estáveis
- Burocrático
 - Sobrecarrega desenvolvimento, pode comprometer a velocidade do projeto
- Possui forte resistência a mudanças

Gerenciamento de Projeto de Software

■ Particularidades

- Invisibilidade
 - Progresso não é imediatamente visível
- Complexidade
- Flexibilidade
 - Propenso a um alto grau de mudança
- Dificuldade de antever suas funcionalidades
- Necessidades surgem durante seu desenvolvimento, e vão amadurecendo até a sua implantação

■ A mudança se torna inevitável

Agilidade

- Agilidade

- Rapidez, desembaraço
- Qualidade de quem é veloz

- Capacidade de responder rapidamente a **mudanças**

- Mudanças de tecnologias, de equipe, de requisitos...

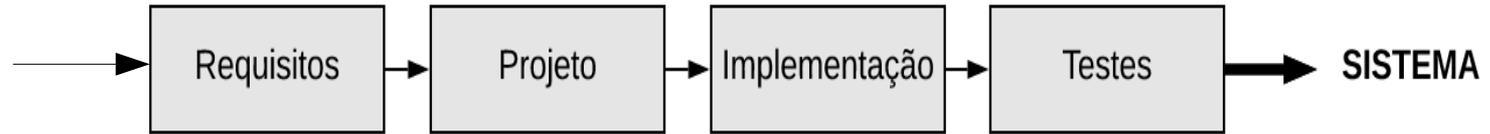
- Entregar valor ao cliente quando se lida com imprevisibilidade e dinamismo dos projetos

- Problema

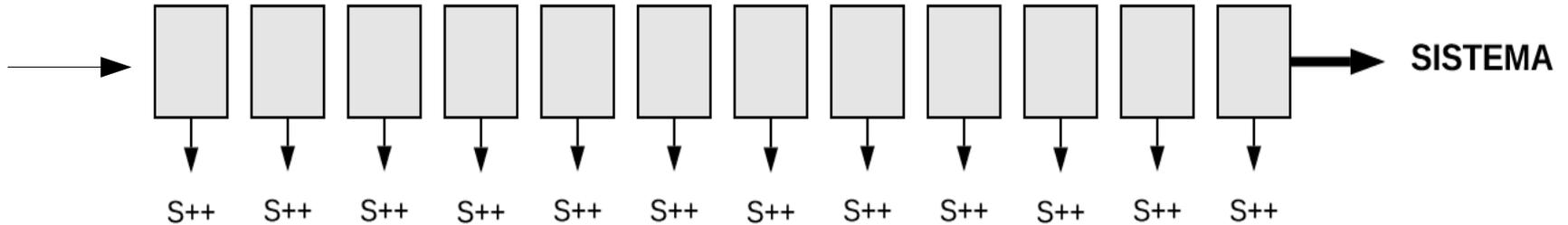
- Aparenta ser indisciplinado

Ideia Central

Tradicional (waterfall)



Ágil

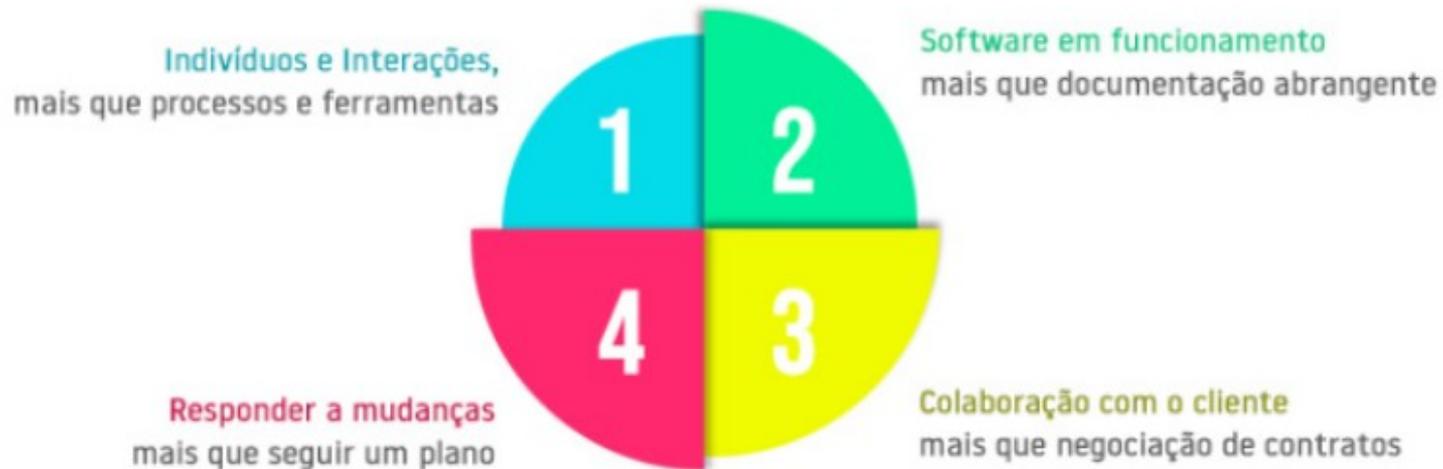


Manifesto Ágil (2001)



Manifesto Ágil

“Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, **passamos a valorizar:**



Princípios do Manifesto Ágil

- Nossa maior prioridade é **satisfazer o cliente** através da entrega contínua e adiantada de software com valor agregado.
- **Mudanças nos requisitos são bem-vindas**, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças visando vantagem competitiva para o cliente.
- **Entregar frequentemente software funcionando**, de poucas semanas a poucos meses, com preferência à menor escala de tempo.
- **Pessoas de negócio e desenvolvedores devem trabalhar diariamente em conjunto** por todo o projeto.

Princípios do Manifesto Ágil

- Construa projetos em torno de **indivíduos motivados**. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
- O método mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de **conversa face a face**.
- **Software funcionando** é a medida primária de progresso.
- **Os processos ágeis promovem desenvolvimento sustentável**. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente.

Princípios do Manifesto Ágil

- Atenção contínua para com a excelência técnica e para com bons projetos aumenta a agilidade.
- Simplicidade –a arte de maximizar o volume de trabalho não realizado é essencial.
- As melhores arquiteturas, requisitos e projetos surgem de equipes auto-organizadas.
- Em intervalos regulares, a equipe se avalia para ver como pode se tornar mais eficiente, então, sintoniza e ajusta seu comportamento de acordo

Busca Ativa

Ler o manifesto na integra:
<http://agilemanifesto.org>



Fonte: <http://www.editoraopet.com.br/blog/importancia-do-incentivo-a-pratica-de-leitura-por-meio-dos-pais-e-educadores/>

Desenvolvimento Ágil de Software

■ Características fundamentais

- Os processos de especificação, projeto e implementação são intercalados.
 - Não há especificação detalhada do sistema, e a documentação do projeto é minimizada ou gerada automaticamente pelo ambiente de programação usado para implementar o sistema.
 - O documento de requisitos do usuário apenas define as características mais importantes do sistema.

Desenvolvimento Ágil de Software

- Desenvolvimento de sistema personalizado dentro de uma organização, em que existe um **compromisso claro do cliente** de se **envolver** no processo de desenvolvimento, e em que não há muitas regras e regulamentos externos que afetam o software.

Praticando...

- Diante do entendimento anterior sobre métodos ágeis, cite alguns problemas que podem ser observados durante sua prática.

Métodos Ágeis: Problemas

- Embora a ideia de envolvimento do cliente no processo de desenvolvimento seja atraente, seu sucesso depende de um **cliente disposto** e capaz de **passar o tempo com a equipe de desenvolvimento**, e que possa representar todos os stakeholders do sistema.
 - Frequentemente, os representantes dos clientes estão sujeitos a diversas pressões e não podem participar plenamente do desenvolvimento de software.

Métodos Ágeis: Problemas

- Membros individuais da equipe podem não ter personalidade adequada para o intenso envolvimento que é típico dos métodos ágeis e, portanto, não interagem bem com outros membros da equipe.
- Priorizar as mudanças pode ser extremamente difícil, especialmente em sistemas nos quais existem muitos stakeholders. Normalmente, cada stakeholder dá prioridade diferente para mudanças diferentes.

Métodos Ágeis: Problemas

- Manter a simplicidade exige um trabalho extra.
- Sob a pressão de cronogramas de entrega, os membros da equipe podem não ter tempo para fazer as simplificações desejáveis.
- Muitas organizações, principalmente as grandes empresas, passaram anos mudando sua cultura para que os processos fossem definidos e seguidos.
- É difícil para eles mudar de um modelo de trabalho em que os processos são informais e definidos pelas equipes de desenvolvimento.

Metodologias Ágeis

- Scrum
- Kanban
- Extreme programming (XP)
- Lean

Observação Importante

- Todo método de desenvolvimento deve ser entendido como um conjunto de recomendações.
 - Cabe a uma organização analisar cada uma e decidir se ela faz sentido no seu contexto

Pontos Importantes

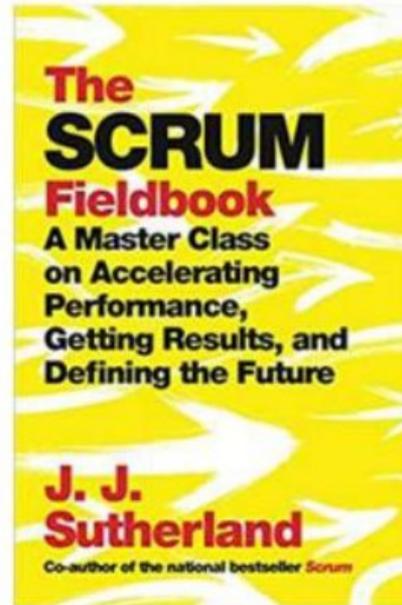
- Nenhum processo é uma receita de bolo
- Processo ajuda a não cometer certos "grandes erros"
- Processos não são adotados 100% igual ao manual
- Bom senso é importante
- Experimentação é importante

Conceito de Scrum

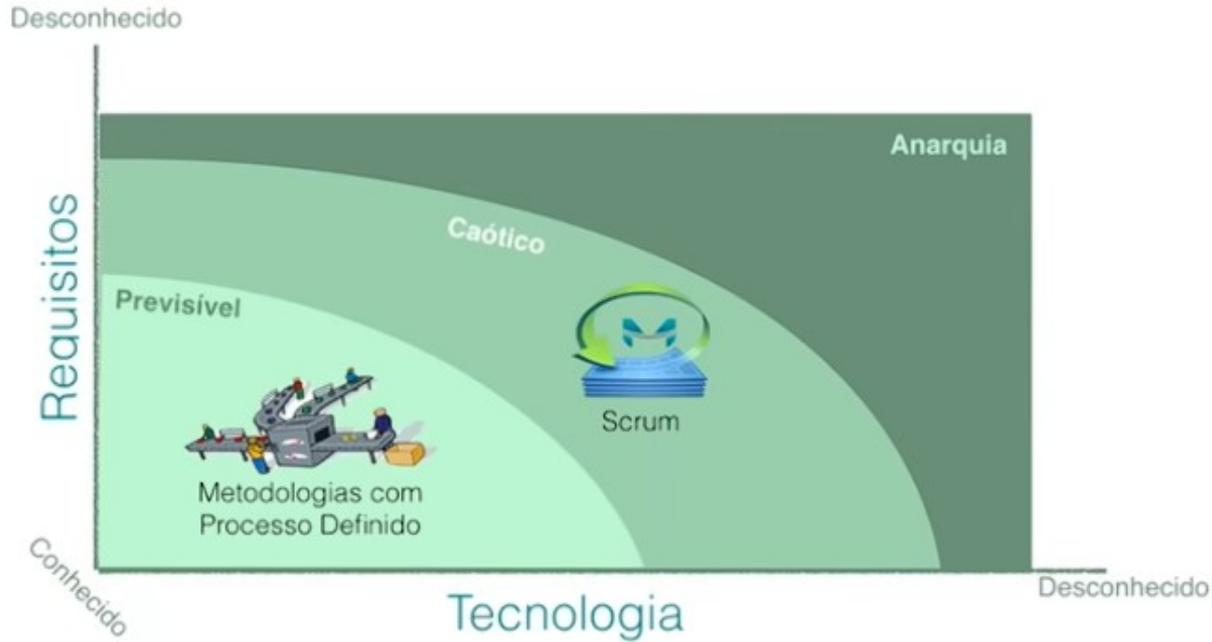
- É um conjunto de ações para gestão de projetos de alta performance ou complexidade, de forma colaborativa, incremental e com otimização de recursos.
- É um framework **simples** para gerenciar projetos **complexos**.

Scrum

- Scrum é uma indústria: livros, consultoria, certificações, marketing



Entendendo Melhor...



Scrum

“O framework Scrum é um conjunto de valores, princípios e práticas que fornecem a base para que a sua organização adicione suas práticas particulares de engenharia e gestão e que sejam relevantes para a realidade da sua empresa. O resultado será uma versão de Scrum que é **exclusivamente sua.**”

Pilares



Transparência
dos processos,
requisitos de
entrega e status

Transparência



Inspeção
constante de
tudo o que está
sendo feito

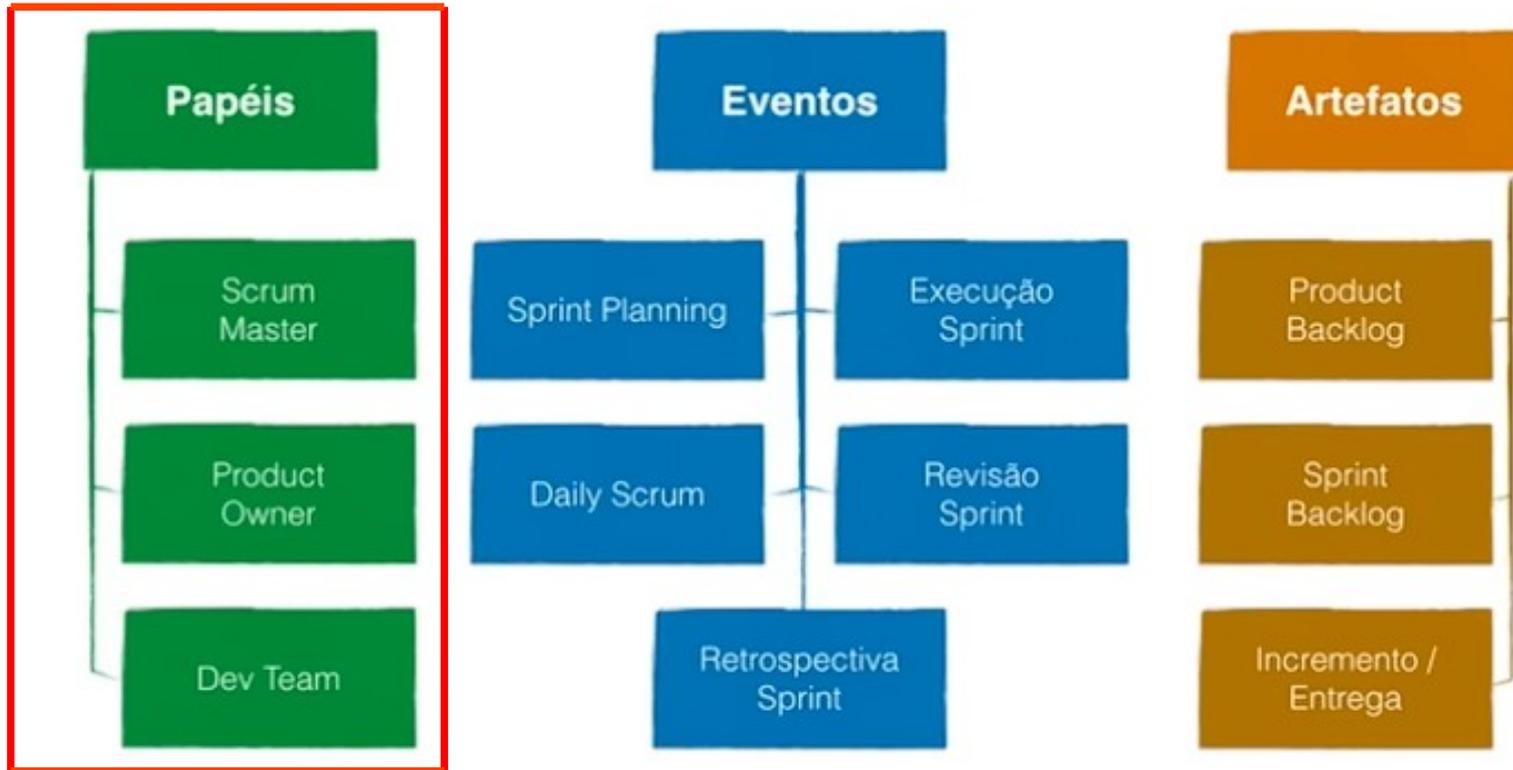
Inspeção



Adaptação, tanto
do processo,
quanto do produto
às mudanças

Adaptação

Componentes do Scrum



TIME SCRUM

TIME SCRUM

Time Scrum

- O Time SCRUM é um conjunto multi-disciplinar de profissionais. Estes times devem ser autogeridos, ou seja, não existe hierarquia ou a figura de um líder. O time deve decidir, em conjunto a melhor forma de completar o seu trabalho ao invés de receber ordens externas de alguém que não faz parte da equipe e por isto não tem o conhecimento empírico para determinar prazos, por exemplo. A opinião de todos deve ser igualmente ouvida e respeitada. Mas existem alguns papéis fixos para auxiliar o bom andamento do trabalho.

- **PRODUCT OWNER**
- **SCRUM MASTER**
- **DEVELOPMENT TEAM**



Time Scrum

■ **PRODUCT OWNER** (DONO DO PRODUTO, PROPRIETÁRIO DO PRODUTO)

- A responsabilidade do Product Owner, ou Dono do Produto, é garantir a qualidade do trabalho. Este cara representa a voz do cliente, do dono da empresa, dos investidores, enfim, de todas as pessoas externas a equipe... Ele é responsável por traduzir as histórias dos usuários em uma lista de tarefas concretas, expressar claramente quais são as funcionalidades a serem desenvolvidas e definir a prioridade de cada uma de uma forma objetiva.



Responsável por

- Definir a Visão do Produto
- Elaborar e manter o Product Backlog;
- Definir a prioridade e ROI;
- Representar o cliente;
- Aceitar ou rejeitar os entregáveis.

Dica: Para desenvolvimento comercial, o Product Owner pode ser o Gerente de Produto. Para esforços internos de desenvolvimento, o Product Owner poderia ser o gerente da função de negócios em que se está trabalhando.

Product Owner

- Outras responsabilidades
 - Escrever histórias dos usuários
 - Explicar histórias para os devs, durante o sprint
 - Definir "testes de aceitação" de histórias
 - Priorizar histórias

Time Scrum

■ SCRUM MASTER (MESTRE SCRUM)

- O SCRUM Master, ao contrário do que o título pode sugerir, não manda na equipe. Muito pelo contrário, o SCRUM Master serve a equipe. A função dele é facilitar o bom andamento dos jobs e garantir que todos tenham as ferramentas necessárias para cumprir suas tarefas. Tipicamente o SCRUM Master é o cara que conhece melhor as regras do SCRUM e portanto consegue organizar o jogo de maneira mais efetiva. Ele é uma mistura de professor, gerente e ombro amigo.



Responsável por

- Ser um líder (servidor);
- Remover impedimentos;
- Proteger a equipe;
- Ajudar o PO (com Product Backlog);
- Ser o facilitador da equipe;
- Garantir as práticas SCRUM.

Scrum Master

■ Scrum master ajuda o PO

- Encontrando técnicas para o gerenciamento efetivo do Backlog do Produto;
- Claramente comunicar a visão, objetivo e itens do Backlog do Produto para o Time de Desenvolvimento;
- Ensinar a Time Scrum a criar itens de Backlog do Produto de forma clara e concisa;
- Compreender a longo-prazo o planejamento do Produto no ambiente empírico;
- Compreender e praticar a agilidade;
- Facilitar os eventos Scrum conforme exigidos ou necessários.

Scrum Master

■ Scrum master ajuda o time

- Treinar o Time de Desenvolvimento em autogerenciamento e interdisciplinaridade;
- Ensinar e liderar o Time de Desenvolvimento na criação de produtos de alto valor;
- Remover impedimentos para o progresso do Time de Desenvolvimento;
- Facilitar os eventos Scrum conforme exigidos ou necessários;
- Treinar o Time de Desenvolvimento em ambientes organizacionais nos quais o Scrum não é totalmente adotado e compreendido.

Scrum Master

- **Scrum master ajuda a organização:**
 - Liderando e treinando a organização na adoção do Scrum;
 - Planejando implementações Scrum dentro da organização;
 - Ajudando funcionários e partes interessadas a compreender e tornar aplicável o Scrum e o desenvolvimento de produto empírico;
 - Causando mudanças que aumentam a produtividade do Time Scrum;
 - Trabalhando com outros Scrum Masters para aumentar a eficácia da aplicação do Scrum nas organizações.

Time Scrum

■ TEAM (EQUIPE)

- Um pequeno grupo de pessoas (os tradicionalistas sugerem de 4 a 9) com cargos diferentes que vai trabalhar junto para cumprir as metas estabelecidas. É a galera que vai por a mão na massa e entregar um produto “pronto” ao final do ciclo de desenvolvimento. No caso de desenvolvimento web teríamos designers, programadores, front-enders, profissionais de UX, etc. Não importa o seu cargo todos tem o mesmo título: desenvolvedor.



Responsável por

- Fazer estimativa;
- Definir as tarefas;
- Desenvolver o produto;
- Garantir a qualidade do produto;
- Apresentar o produto ao cliente

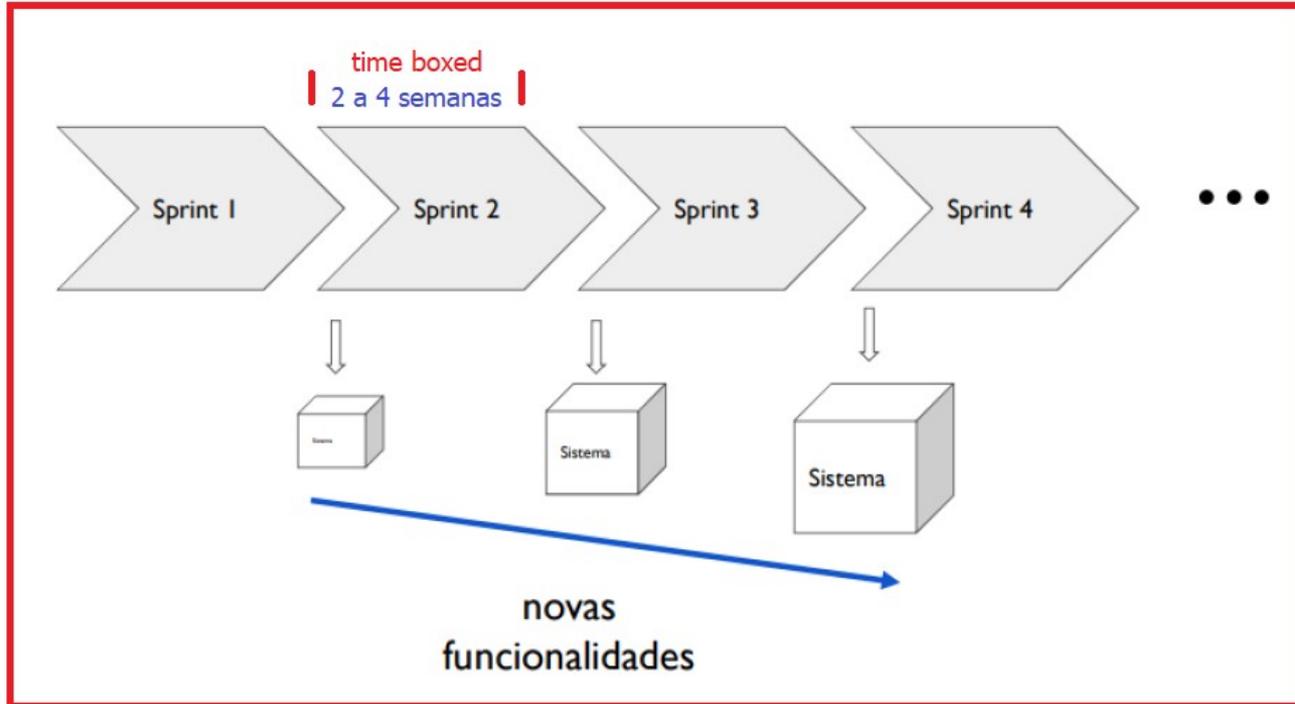
Equipe: auto-gerenciável e multifunciona ISCRUM

Equipe de Desenvolvimento (Dev Team)

Desenvolvimento Tradicional	Desenvolvimento Scrum
<p>No desenvolvimento tradicional de software são abordados vários tipos de trabalho, tais como: arquiteto, programador, testador, administrador de banco de dados, Designer, e assim por diante.</p>	<p>é definido o papel do <u>Time de Desenvolvimento</u>, que é simplesmente a junção de todas essas pessoas em uma equipe multidisciplinar, e que são responsáveis pela concepção, construção e testes do produto.</p>

EVENTOS

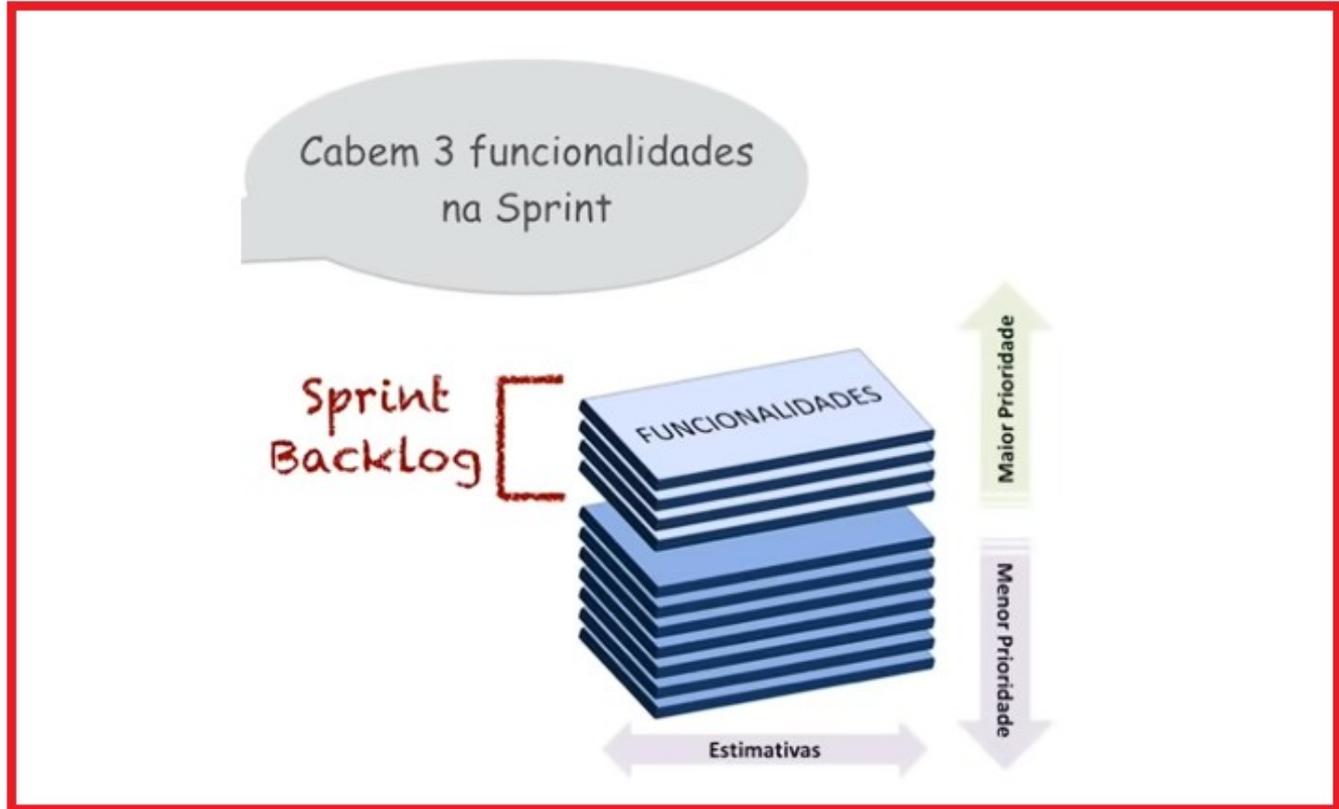
Sprints



Sprint

- Se você puder lembrar apenas de um termo do SCRUM lembre-se deste: Sprint. As pessoas comuns dividem o tempo em dias, meses, horas... Os adeptos do SCRUM dividem o tempo em ciclos de desenvolvimento chamados Sprints. Basicamente uma Sprint é período de tempo que você determinou para cumprir uma determinada tarefa, uma corrida para atingir uma meta. A idéia é ter algo concreto e funcional no final da Sprint. Estes ciclos devem ser períodos de tempo curtos e fixos. Você estipula anteriormente a duração das Sprints (recomenda-se de uma semana a 1 mês). Uma sprint deve começar imediatamente após o termino de outra.
- Durante a Sprint não é possível alterar os objetivos estabelecidos para aquele ciclo ou os profissionais integrantes da equipe. Isto é super importante para garantir a produtividade. Mas se os objetivos tornarem-se obsoletos o Product Owner pode simplesmente cancelar a Sprint.
- Sprints são divididas em etapas menores: reunião de planejamento, reuniões diárias, o trabalho de desenvolvimento em si, revisão e retrospectiva.

Planejamento da Sprint



Planejamento da Sprint

- Nesta reunião serão definidos todos os objetivos da Sprint. Nas regras oficiais, considerando um sprint de 1 mês, esta reunião tem a duração de 8 horas. Você pode adaptar isto proporcionalmente de acordo com o seu ciclo de Sprint. Se o seu Sprint é de 1 semana, por exemplo, esta reunião terá a duração máxima de 2 horas. Em cada metade da reunião duas perguntas devem ser respondidas:
 - - O que será entregue como resultado?
 - Como o trabalho necessário para entregar o produto será realizado?

Sprint: O que se faz?

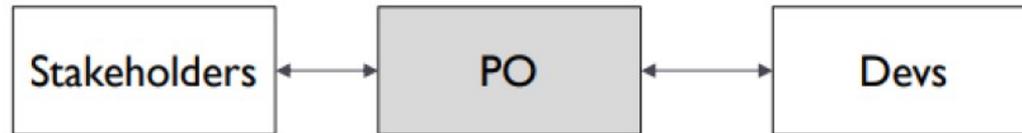
- Implementa-se algumas histórias dos usuários, **escritas pelo PO**;
 - Histórias = funcionalidades (ou features) do sistema:
Exemplo: realizar saque em caixa eletrônico:

- ▶ COMO cliente
- ▶ GOSTARIA de sacar dinheiro no caixa eletrônico
- ▶ POIS ASSIM evitarei a fila do banco

Bem simples, deve caber em um post-it

Sprint: O que se faz?

- Durante sprint, PO explica histórias (requisitos) para devs
- Troca-se documentação formal e escrita por documentação verbal e informal
- Conversas entre PO e Devs



Reunião Diária (Daily Scrum)

- A idéia da reunião diária, ou stand-up meeting, é juntar a equipe para um bate-papo de 15 minutos no máximo para revisar o andamento do projeto. Cada membro da equipe deve responder as seguintes perguntas:
 - - **O que eu consegui completar ontem?**
 - - **O que farei hoje?**
 - - **Quais obstáculos estão impedindo o meu progresso?**
- A função do **SCRUM Master** é moderar a dinâmica e tentar ao máximo solucionar os obstáculos apresentados para que a Sprint seja bem sucedida.
- Ah, e se você está se perguntando o por que do nome é que a recomendação é que estas reuniões sejam realizadas com todos de pé (do inglês stand-up). Desta maneira a reunião não se prolonga por muito tempo e todos ficam mais focados.

Sprint Review

- A idéia aqui, obviamente, é revisar todos os itens desenvolvidos e demonstrar o produto final. A duração desta reunião considerando um sprint de um mês é 4 horas.

Retrospectiva

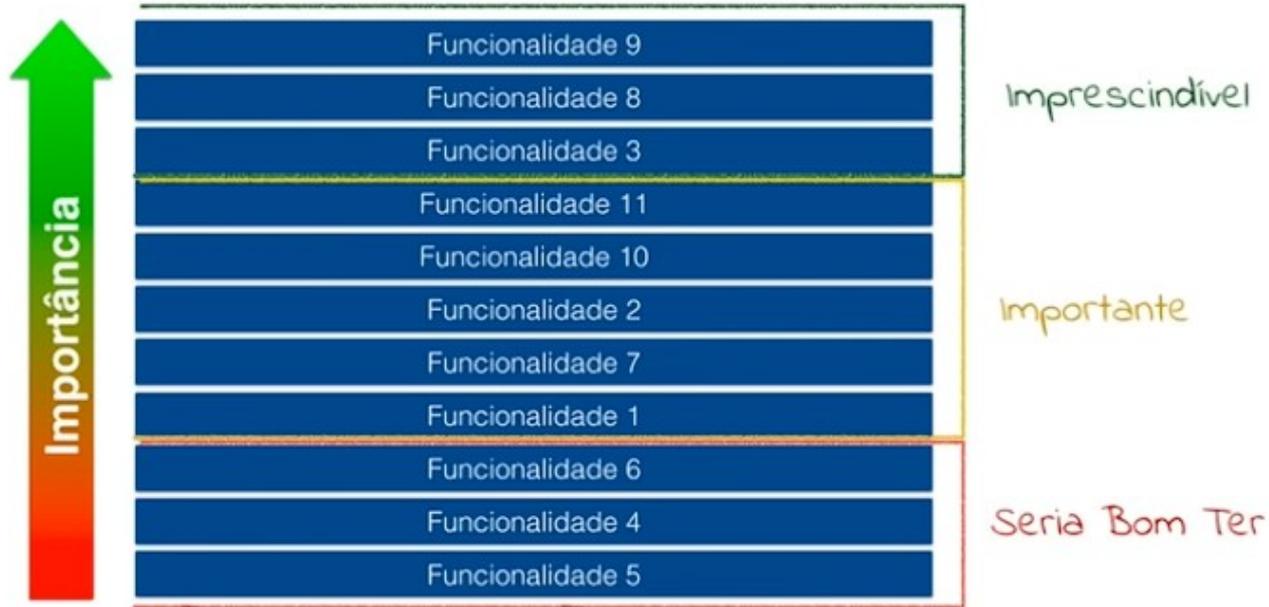
- A sprint acabou! Agora é hora de olhar para trás e repensar o que deu certo, o que deu errado e planejar o que pode ser melhorado no futuro.

ARTEFATOS

Product Backlog

- O **Product Backlog** é uma compilação de tudo o que o seu cliente gostaria de realizar no projeto. Pense com uma grande **wishlist** com todos os recursos e funcionalidades que os usuários gostariam de ver presentes no trabalho final.
- Esta lista deve ser organizada pelo **Product Owner** de acordo com valor, risco, prioridade e necessidade. Por natureza o backlog é algo dinâmico e muda constantemente para incluir as novas solicitações dos clientes, usuários e do mercado em si.
- Desta forma os produtos são sempre aperfeiçoados em cada Sprint.

Product Backlog



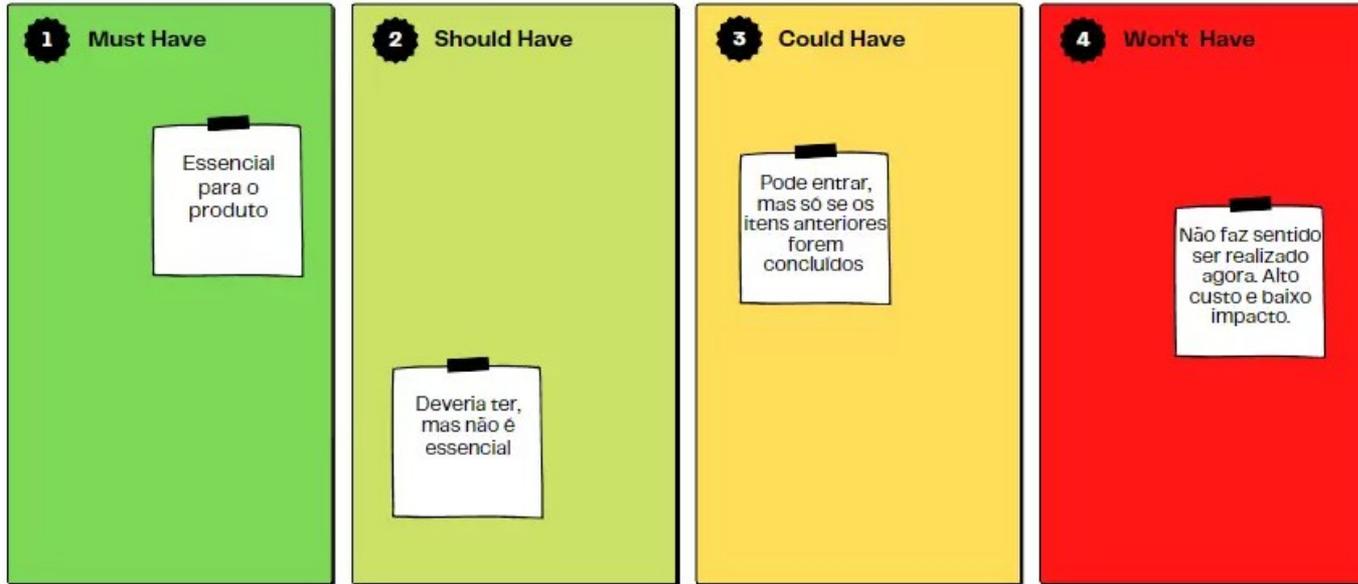
PRODUCT BACKLOG

Backlog do Produto

- Médico ou garçom?
 - Em um restaurante: garçom mostra o cardápio para você escolher o que deseja;
 - Médico pede exames ou te examinará. Ou seja, ele entenderá sua dor para, depois, pensar em remédios, tratamentos ou soluções para o seu problema
 - *Pense como médico, não como garçom.*

Técnicas de Priorização

- Método MoSCoW



Método MoSCoW

- **Must Have:** “Tem que ser feito” numa tradução literal, seria a categoria para as tarefas mais indispensáveis para o produto, no qual sem elas poderia se considerar um fracasso;
- **Should Have:** “Deveria ter”, ou seja, é importante, mas não crucial, por isso são tarefas que devem vir logo após as categorizadas como essenciais;
- **Could Have:** “Poderia ter”, tarefas desejáveis, mas que também não necessárias, ou seja, a serem priorizadas apenas se as tarefas das categorias anteriores forem completadas;
- **Won't Have:** “Não será feito”, tarefas que envolvem muito esforço e têm baixo impacto. Não devem ser priorizadas no momento.

Exemplo de Product Backlog

PRODUCT BACKLOG				
id	nome	descrição	importancia	tamanho
1	registrar veículo	o dono da garagem precisa do registro do veículo para manter o cadastro dos dados do veículo para futuramente realizar uma venda por procuração	6 	8
2	armazenar veículo	o dono da garagem precisa saber de quem ele comprou e para quem ele vendeu o veículo. Para ter melhor controle dos donos de veículo.	3	5
3	controlar gastos	o funcionário da garagem precisa registrar os gastos com o veículo de todas as naturezas para calcular o preço de venda do veículo.	4	8
4	pagar comissão	o gerente da loja precisa calcular a comissão dos funcionários para pagar o salário do funcionário	7 	2
5	consultar veículos	o funcionário pode consultar o histórico do veículo para saber o antigo proprietário	1	8
6	simular preço	o responsável pelas vendas precisa visualizar o custo total do veículo para saber o preço final da venda.	5	13
7	vender veículo	o responsável pelas vendas precisa registrar a saída do veículo da garagem para controlar as vendas	2	13

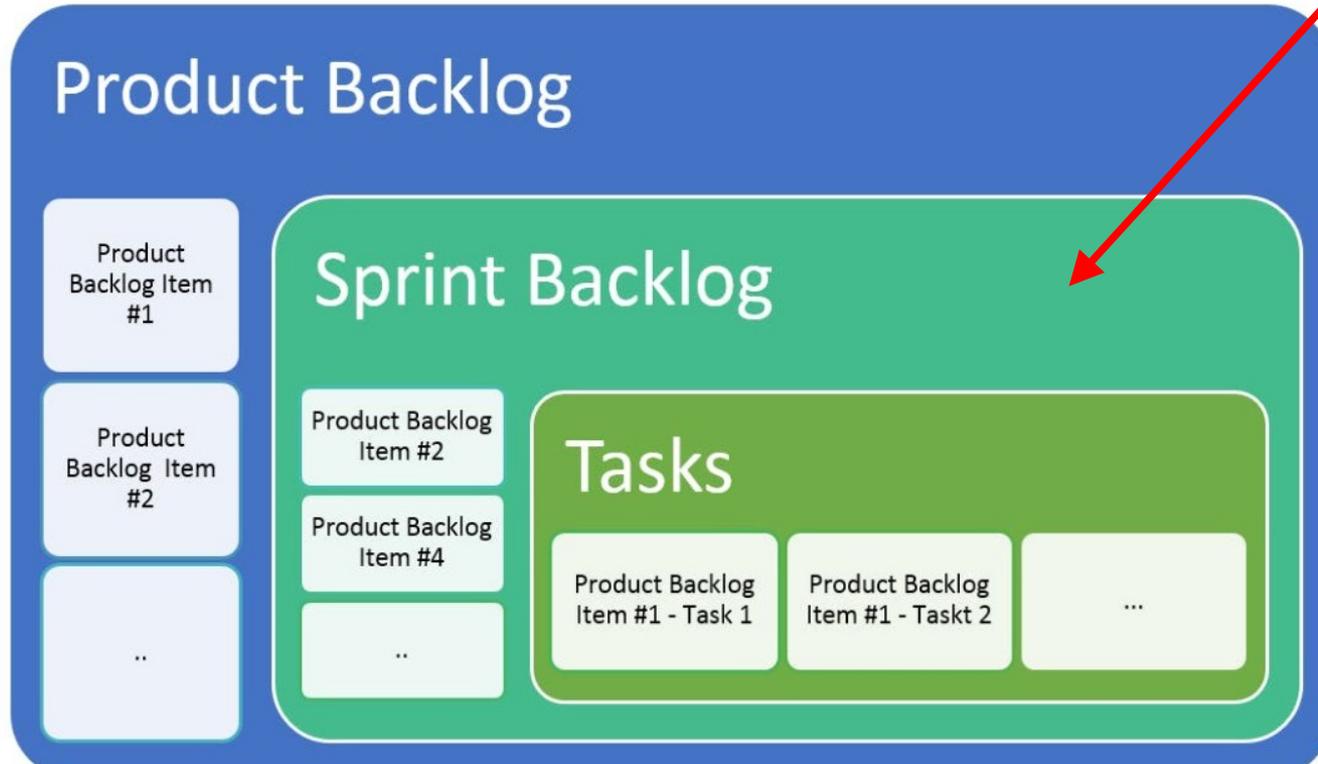
Exemplo de Backlog do Produto

BP01 - Backlog do Produto					
ID	Nome	Importância	PH	Demonstrar	Notas
HU1	Cadastrar as taxas do estacionamento.	30	8	Como um usuário do sistema eu quero cadastrar o valor das taxas de estacionamento (por hora/diária/mensalista e fração por 15min) para realizar o calculo do valor a ser pago posteriormente.	Exigir senha de gerente do sistema.
HU2	Visualizar dados do Estacionamento.	30	8	Como um usuário do sistema eu quero visualizar informações do estacionamento para ver número de vagas livres e ocupadas, taxas do estacionamento por hora/diária/mensalista e fração por 15min.	

Backlog da Sprint

- O **Backlog** da Sprint é um conjunto de itens do Backlog do Produto selecionados para a Sprint, juntamente com o plano de entrega do incremento do produto e atingir o objetivo da Sprint.
- O **Backlog** da Sprint é a previsão da Equipe de Desenvolvimento sobre qual funcionalidade estará no próximo incremento e do trabalho necessário para entregar a funcionalidade.
- O **Backlog** da Sprint define qual trabalho a Equipe de Desenvolvimento realizará para converter os itens do Backlog do Produto em um incremento “Pronto”. O **Backlog** do Produto torna visível todo o trabalho que a Equipe de Desenvolvimento identifica como necessário para atingir o objetivo da Sprint.

Sprint Backlog



Incremento do Produto

- O incremento é a soma de todos os itens do Backlog do Produto completados durante a Sprint e tudo das Sprints anteriores.
- Ao final da Sprint um novo incremento deve estar “Pronto”, o que significa que deve estar na condição utilizável e atender a definição de “Pronto” do Time SCRUM.
- Este deve estar na condição utilizável independente do **Product Owner** decidir por liberá-lo realmente ou não.

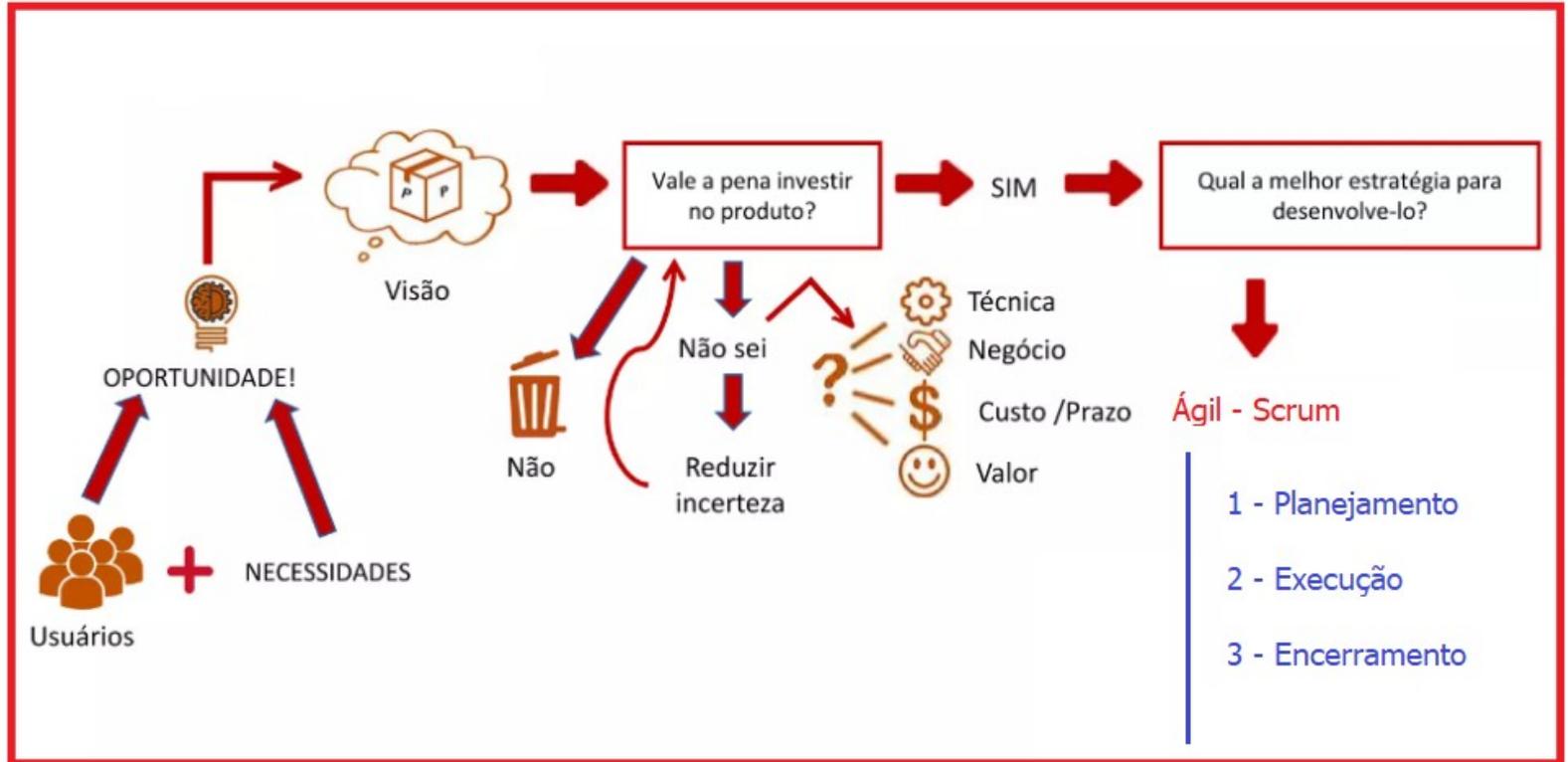
Definição de “Pronto”

- Quando o item do Backlog do Produto ou um incremento é descrito como “Pronto”, todos devem entender o que o “Pronto” significa.
- Embora, isso varie significativamente de um extremo ao outro para cada Time SCRUM, os integrantes devem ter um entendimento compartilhado do que significa o trabalho estar completo, assegurando a transparência.
- Esta é a “Definição de Pronto” para o Time SCRUM e é usado para assegurar quando o trabalho está completado no incremento do produto.

DINÂMICA DO SCRUM

Dinâmica do Scrum

■ Visão do Produto



Dinâmica do Scrum

1- Planejamento:

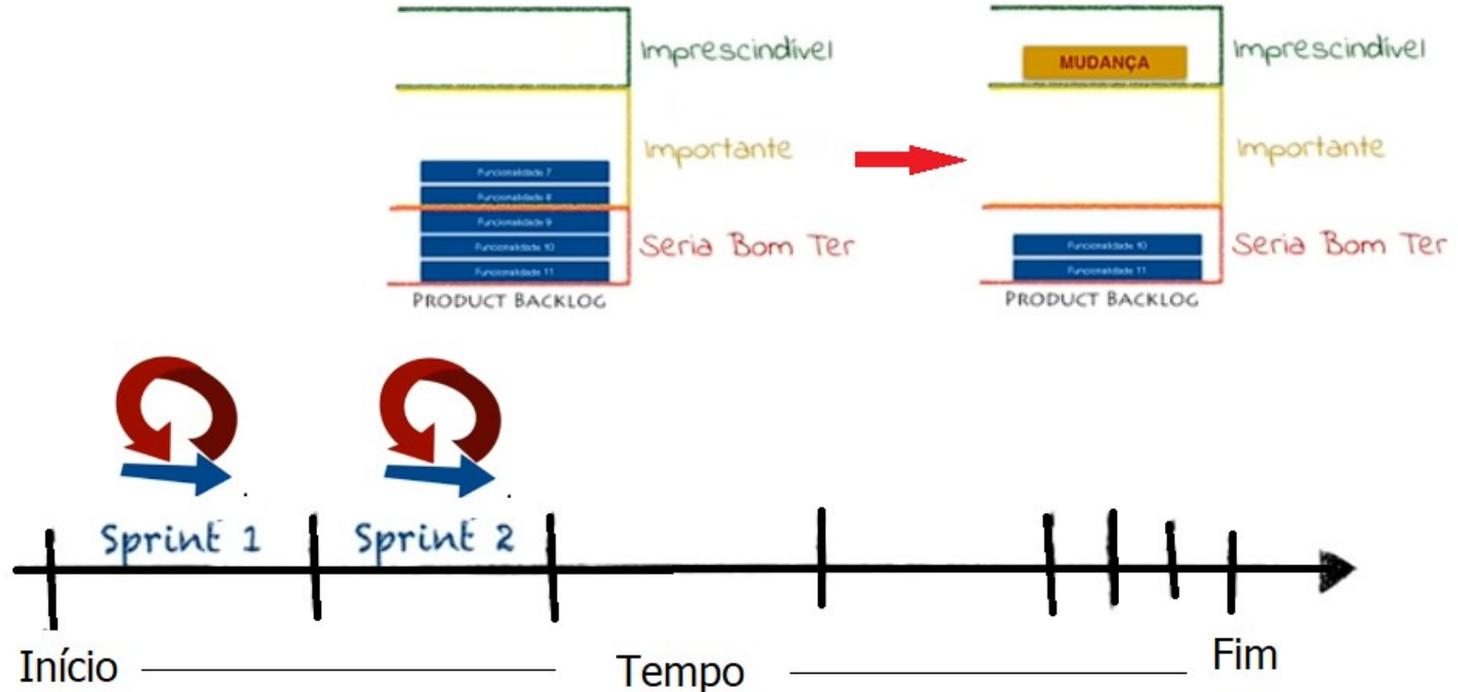
- Definição de novas funcionalidades baseado no conhecimento do sistema como um todo.
- Documento contendo a lista dos requisitos, em ordem de prioridade: Backlog do Produto (**Product Backlog**), que deve ser constantemente atualizado e priorizado.
- Técnicas para obter a lista de requisitos: entrevistas e questionários com o dono do projeto (**Product Owner**).
- Reunião de planejamento (**Sprint Planning Meeting**), visando definir a iteração (*Sprint*) inicial do projeto.
- **Cada *Sprint* tem duração de 2 a 4 semanas.**
- A equipe separa uma parte do topo do *Product Backlog* para o *Sprint*, formando o **Sprint Backlog**.

Dinâmica do Scrum

2- Desenvolvimento:

- As tarefas do *Sprint Backlog* são divididas entre os membros do time que são responsáveis por programar as novas funcionalidades e testá-las, respeitando o tempo previsto, requisitos exigidos e qualidade.
- Ao longo do Sprint reuniões são feitas diariamente (*Scrum Daily Meeting*), no mesmo horário, para acompanhar o progresso do trabalho.

Dinâmica do Scrum: Visão Gráfica



Dinâmica do Scrum

3- Encerramento:

- A saída do Sprint é um conjunto de funcionalidades (incrementos) 100% desenvolvidas.

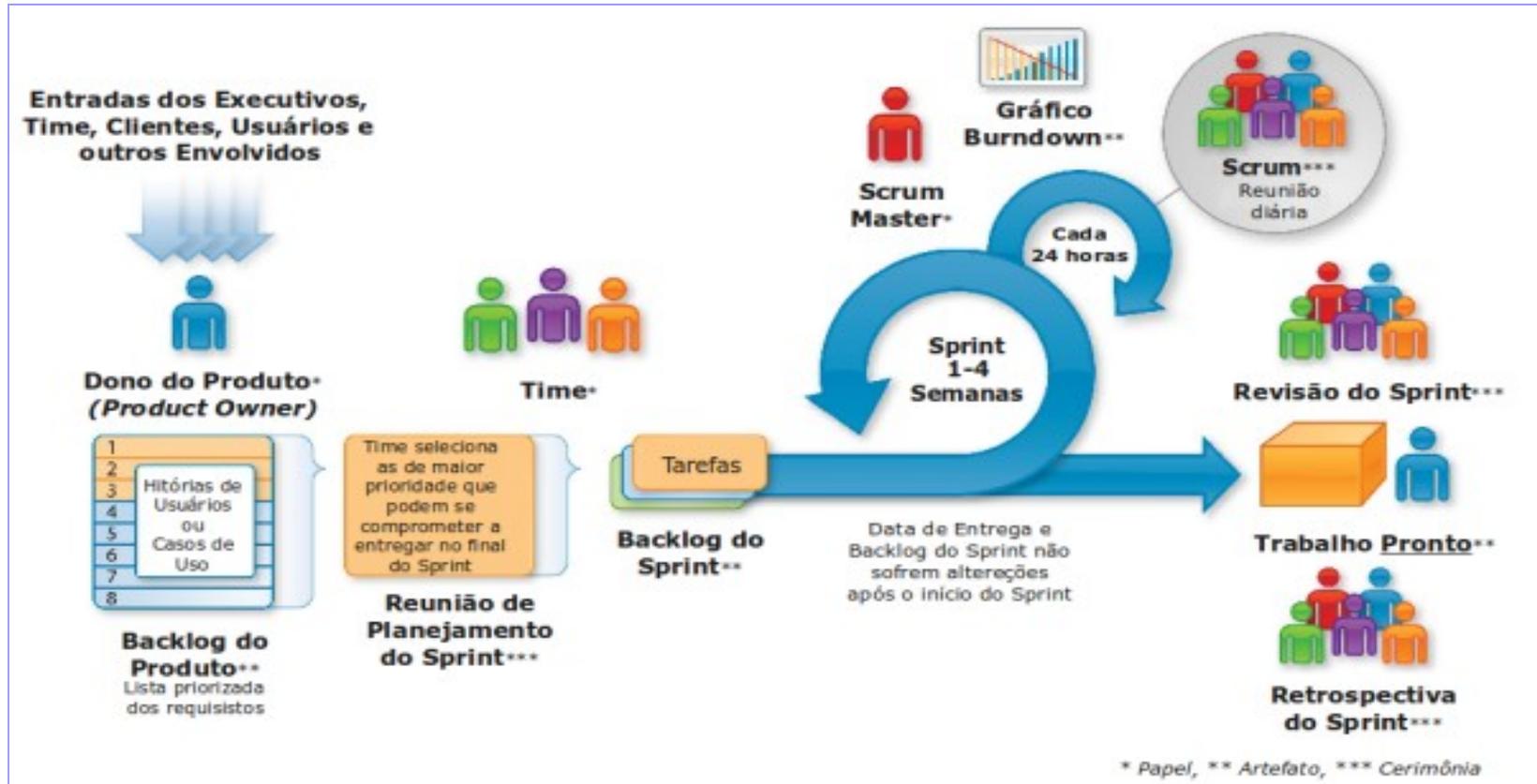
Sprint Review Meeting

Reunião de revisão realizada para que seja apresentado o resultado alcançado. São validadas as funcionalidades e caso sejam necessárias, adaptações são realizadas.

Sprint Retrospective Meeting

Reunião de retrospectiva que tem foco no processo, sendo uma oportunidade para a equipe se inspecionar, visando à criação de um plano de melhorias para os próximos trabalhos.

Fluxo Completo

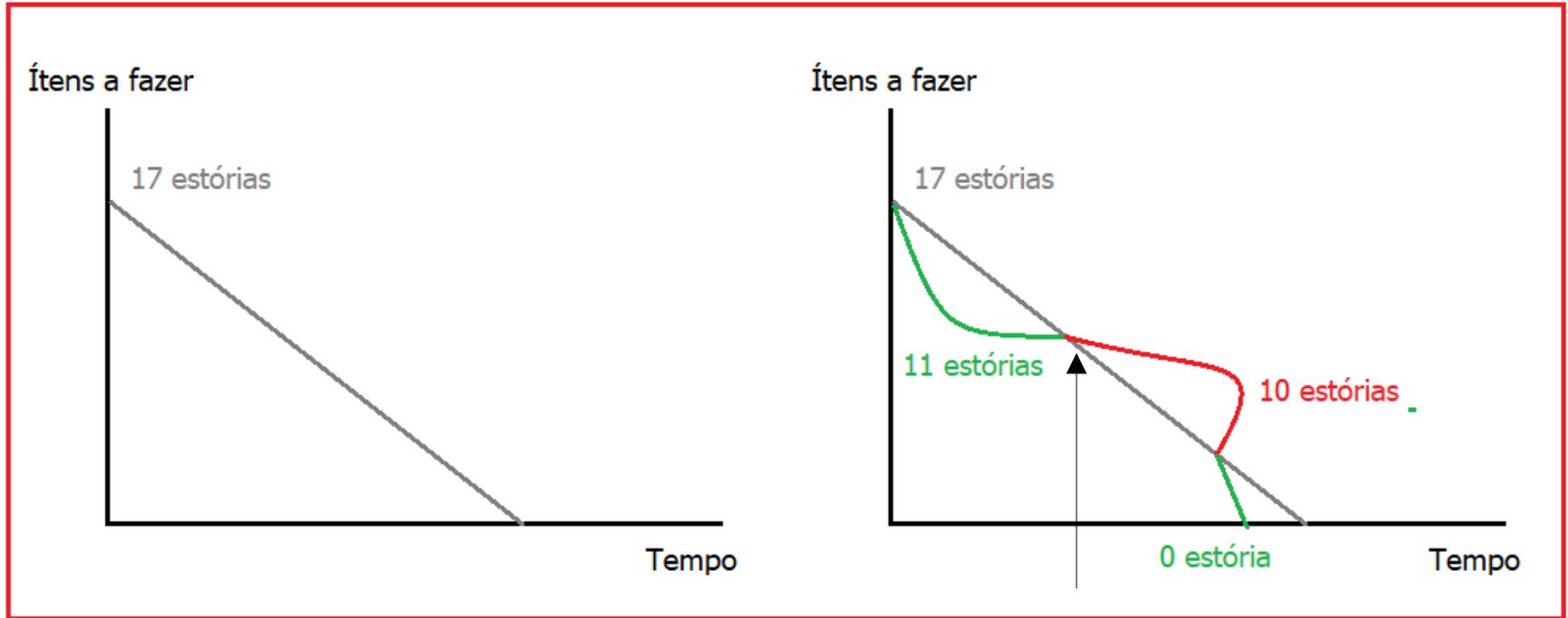


OUTROS CONCEITOS

Burndown Chart

- O ***Burndown*** é um simples gráfico, com dois eixos X e Y, baseado nas atividades que não ultrapassem um dia de trabalho.
- O eixo Y indica o número de tarefas existentes no Sprint e o eixo X os dias que representam o tamanho do Sprint.

Burndown Chart



Kanban Board

Backlog

Fazendo

Validação

Concluído

Cadastro
usuário

Consulta
Cliente

Consulta
Usuário

Mensagens
de
confirmação

Design
Básicos de
Telas

Tela de
Enc

Validação
CPF

Tela Alterar
Senha

Erro Login

Erro Tela
Produtos

Melhoria
Performance

Erro
Consulta
CEP

Quem Utiliza Scrum

Google™



BBC



SIEMENS

YAHOO!

Borland



Microsoft™

PHILIPS



NOKIA
Connecting People

nielsen
.....

Canon

THE DOCUMENT COMPANY
XEROX®



salesforce.com ~~OPEN~~
Success On Demand.™

Scrum Solo

- SCRUM é baseado em pequenas equipes.
- Ele permite a comunicação entre os membros da equipe.
- Entretanto, há uma grande quantidade de softwares desenvolvidos por programadores solos.
- Um software sendo desenvolvido por um só programador pode ainda se beneficiar de alguns princípios do SCRUM, como: um *backlog* de produto, um *backlog* de *sprint*, um *sprint* e uma retrospectiva de *sprint*. SCRUM Solo é uma versão adaptada para uso de programadores solo.

GQM (*Goal Question Metric*)

- Como uma das funções do SCRUM Master é zelar pelo processo, a medição deve ser um aliado importante para seu trabalho.
- Uma boa prática seria analisar os pontos negativos e positivos levantados nas retrospectivas (principalmente os recorrentes) e identificar pontos falhos no processo.
- Por exemplo, um ponto negativo recorrente nas retrospectivas pode ser que muitos bugs estão sendo encontrados na versão de homologação, mostrando que a atividade de testes automatizados do processo de trabalho está falha.

GQM (Goal Question Metric)

- A partir desta identificação, é possível aplicar o GQM para planejar o processo de medição, isto ocorre definindo:
 - O objetivo da medição (GOAL).
 - Ex: Melhorar a eficiência dos testes automatizados.
 - A questão a ser respondida (QUESTION).
 - Ex: Qual a eficiência dos testes automatizados?
 - As métricas que respondem a questão respondida (METRIC).
 - Ex: Intervalo entre falhas, Número de bugs encontrados em homologação etc.