

METODOLOGIA ÁGIL

O que é Software?

- São instruções (programas de computadores) que, quando executadas, fornecem as características, funções e desempenho desejados.
- Estruturas de dados que permitem aos programas manipular adequadamente a informação.
- (Sommerville) Softwares são programas de computador e documentação associada. Produtos de software podem ser desenvolvidos para um cliente específico ou para o mercado em geral.

O que é Engenharia de Software?

- Segundo o IEEE, engenharia de software é a aplicação de uma abordagem sistemática, disciplinada e quantificável ao desenvolvimento, operação e manutenção do software.
- Segundo Sommerville, "é uma disciplina de engenharia relacionada com todos os aspectos da produção de software, desde os estágios iniciais de especificação de sistemas até a manutenção desse sistema. Estruturas de dados que permitem aos programas manipular adequadamente a informação. Documentos que descrevem a operação e o uso dos programas.

Modelos de processo de software



Modelos de processo de software

- Representações simplificada de um processo de software
- Representam frameworks
- Modelos mais comuns:
 - Cascata
 - Incremental
 - Orientado a reuso

Figura 2.1

O modelo em cascata

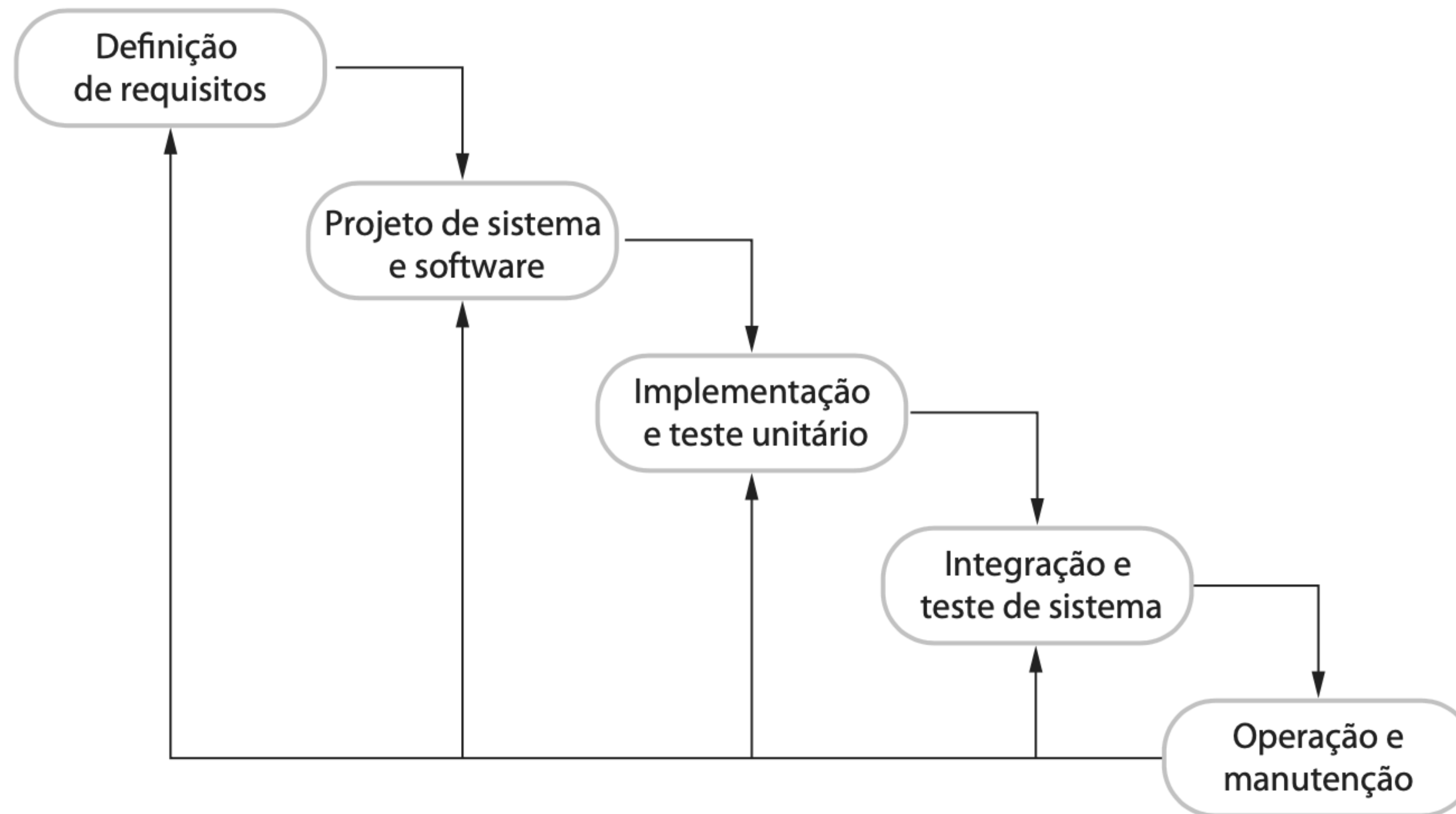
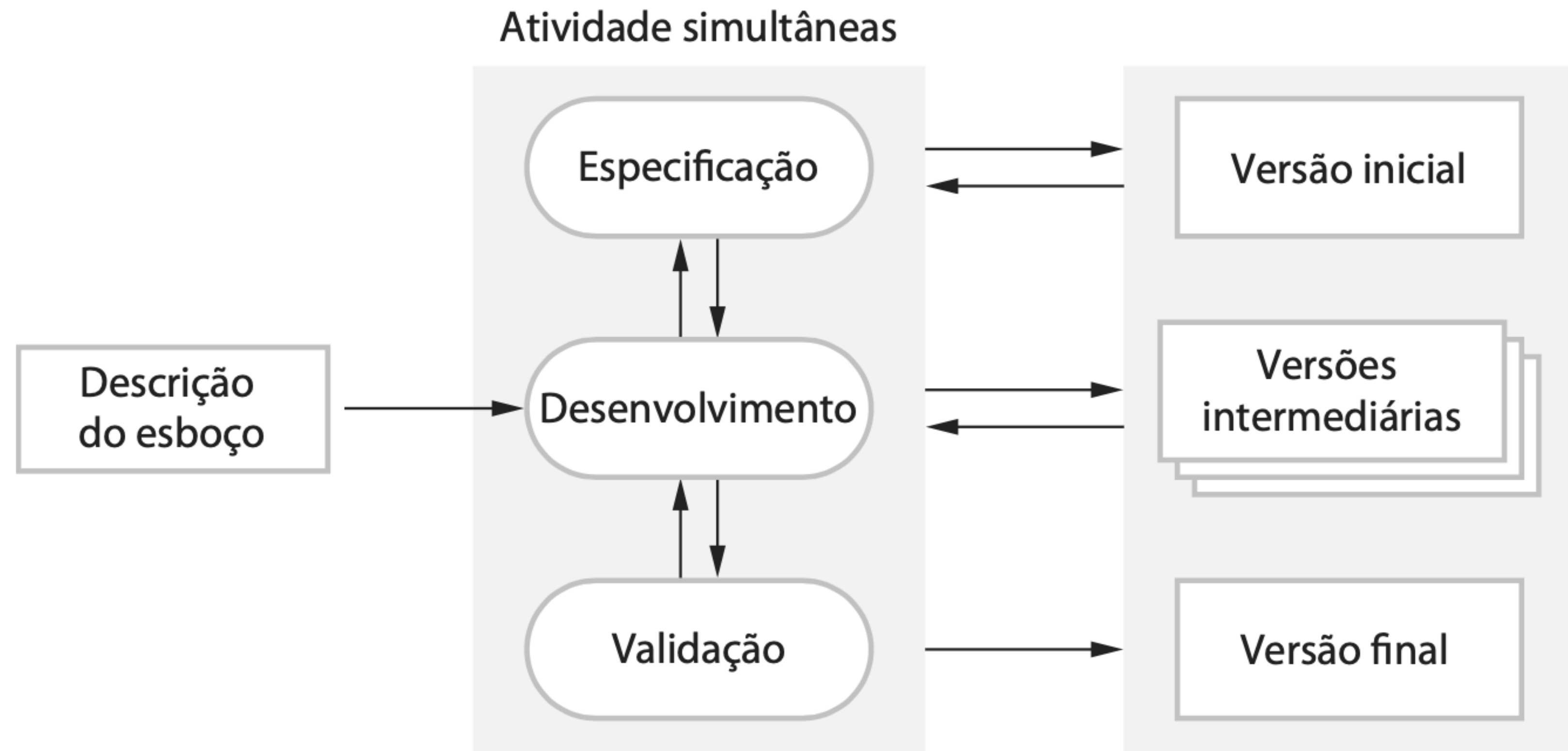


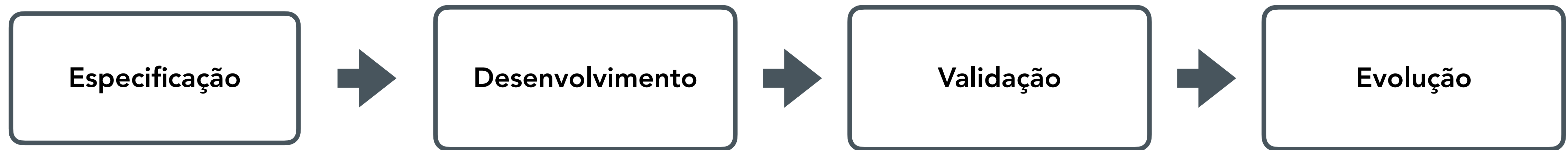
Figura 2.2

Desenvolvimento incremental



Fonte: Sommerville

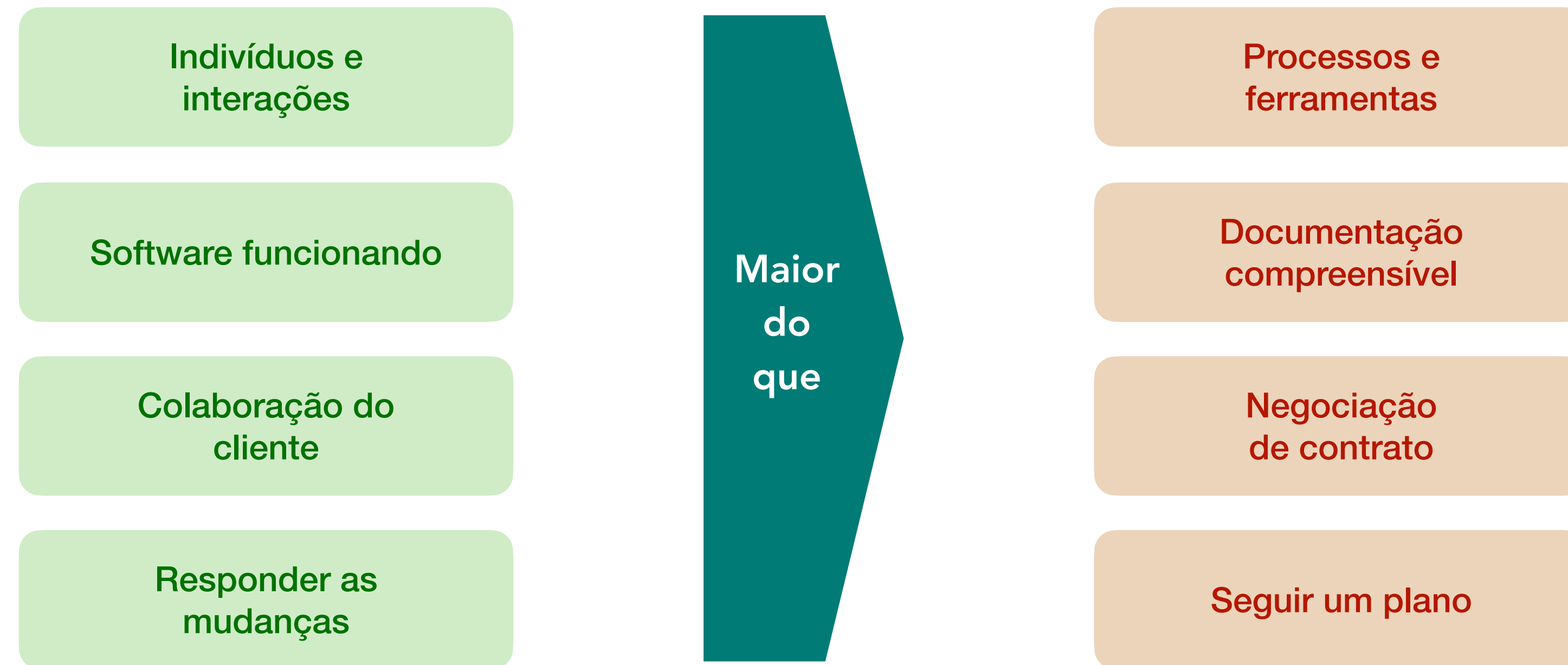
Atividades do processo



Desenvolvimento ágil de software

- Modelo Tradicional (1980's/1990's)
 - Softwares grandes e críticos
 - Sistemas duradouros
 - Grandes empresas
- Modelo Ágil (2000's/2010's)
 - Mudanças rápidas
 - Novos produtos, funções etc
 - Tempo > Qualidade

Manifesto ágil



“Embora os itens à direita sejam importantes,
valorizamos mais os que estão à esquerda”

Fonte: <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>

Metodologias

- Kanban
- Scrum
- Lean
- EXtreme Programming (XP)
- Test Driven Development (TDD)

Scrum

O uso dessa metodologia Scrum, conforme Pressman (2016), vem de uma formação que ocorre durante uma partida de rugby. Essa formação ocorre após uma parada ou quando a bola sai de campo, e o técnico reúne a equipe de jogadores.

A metodologia Scrum foi desenvolvida em 1990 por Jeff Sutherland e Ken Schwabe e, segundo eles, Scrum é um framework para desenvolver e manter produtos complexos. Uma metodologia ágil de software que concentra as atenções no produto final com um rápido desenvolvimento e nas interações dos indivíduos.

Essa metodologia segue os princípios do Manifesto Ágil e conforme Pressman (2016, pg, 78) "são usados para orientar as atividades de desenvolvimento dentro de um processo que incorpora as seguintes atividades metodológicas: requisitos, análise, projeto, evolução e entrega".

Sobre o Scrum

Scrum não é uma metodologia:

Não é um processo padronizado, que metodicamente segue uma série de etapas sequenciais, que garanta a produção, no prazo e no orçamento, de um produto de alta qualidade e que encanta os clientes.

Scrum é um framework:

Para organizar e gerenciar trabalhos complexos, tal como projetos de desenvolvimento de software.

Framework: um conjunto de valores, princípios e práticas que fornecem a base para que a organização adicione suas práticas particulares de gestão, e que sejam relevantes para a realidade da empresa.

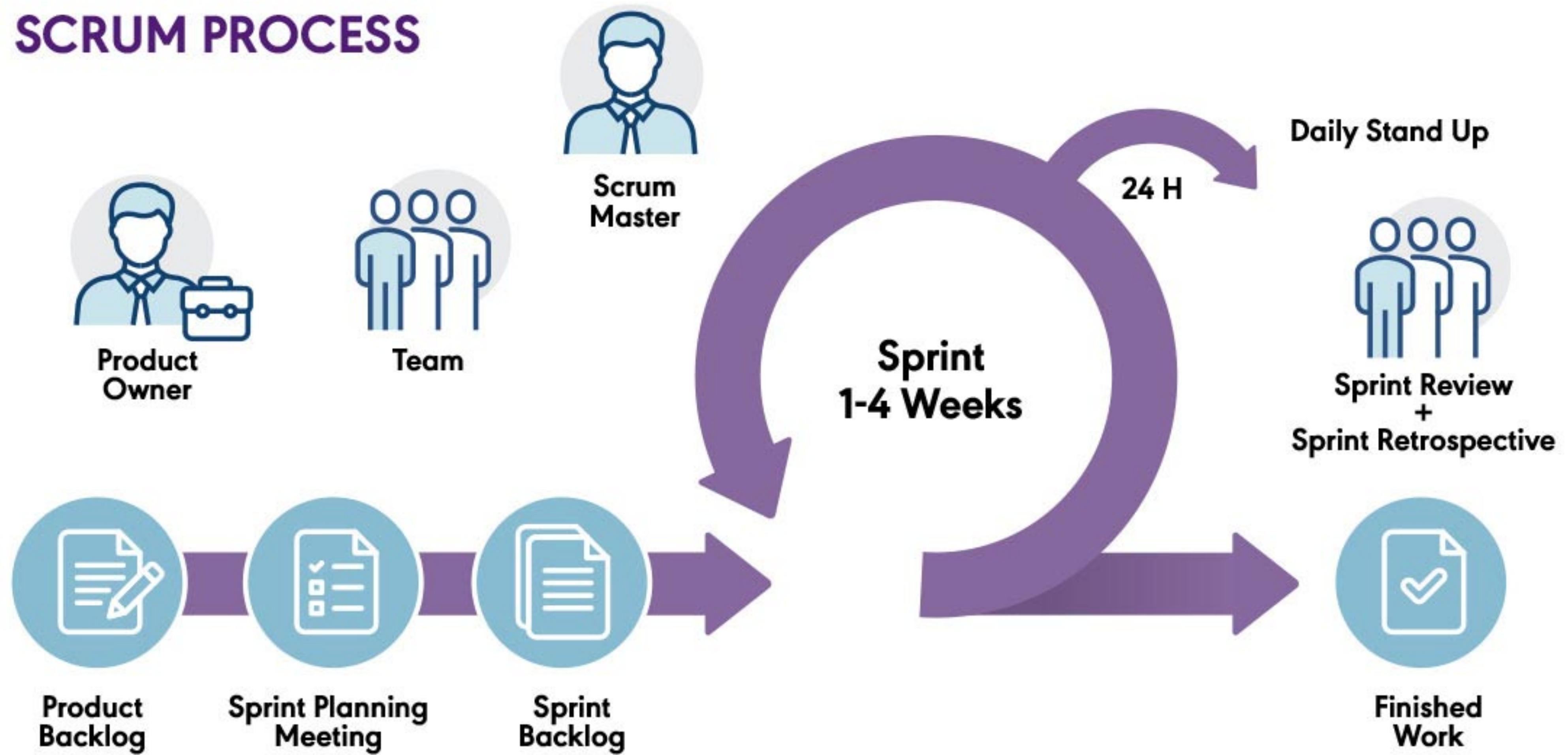
Fundamentos do Scrum

Conforme Sbrocco (2012, pg. 161) sobre o Scrum ser “conhecido pelo fato de estruturar seu funcionamento por ciclos, chamados de sprints que representam iterações de trabalho com duração variável”. As tarefas desenvolvidas dentro de um sprint são adaptadas ao problema pela equipe Scrum e são realizadas em um período curto de tempo enquanto o projeto se desenvolve. Não pode ocorrer nenhuma mudança durante uma sprint, pois o produto é projetado, codificado e testado durante a sprint. O Scrum também estipula um conjunto de práticas e regras que devem ser seguidas pela equipe. No Scrum temos três papéis importantes: **Product Owner, Scrum master e Team.**

No Scrum também ocorrem às cerimônias e são divididas em quatro: **Daily Meeting ou Daily, a Sprint Review, o Sprint Planning e Sprint Retrospective.** A partir dessas cerimônias são gerados os artefatos que os documentos **Product Backlog, Sprint Backlog e o Burndown Chart.**

O Scrum reconhece como princípio básico que os clientes podem mudar de ideia durante o desenvolvimento do projeto e, por isso, adota uma abordagem empírica, ou seja, aceitar que o problema do cliente pode não ser totalmente entendido, mostrando atitude.

SCRUM PROCESS



Papéis do Scrum: Product Owner (PO)

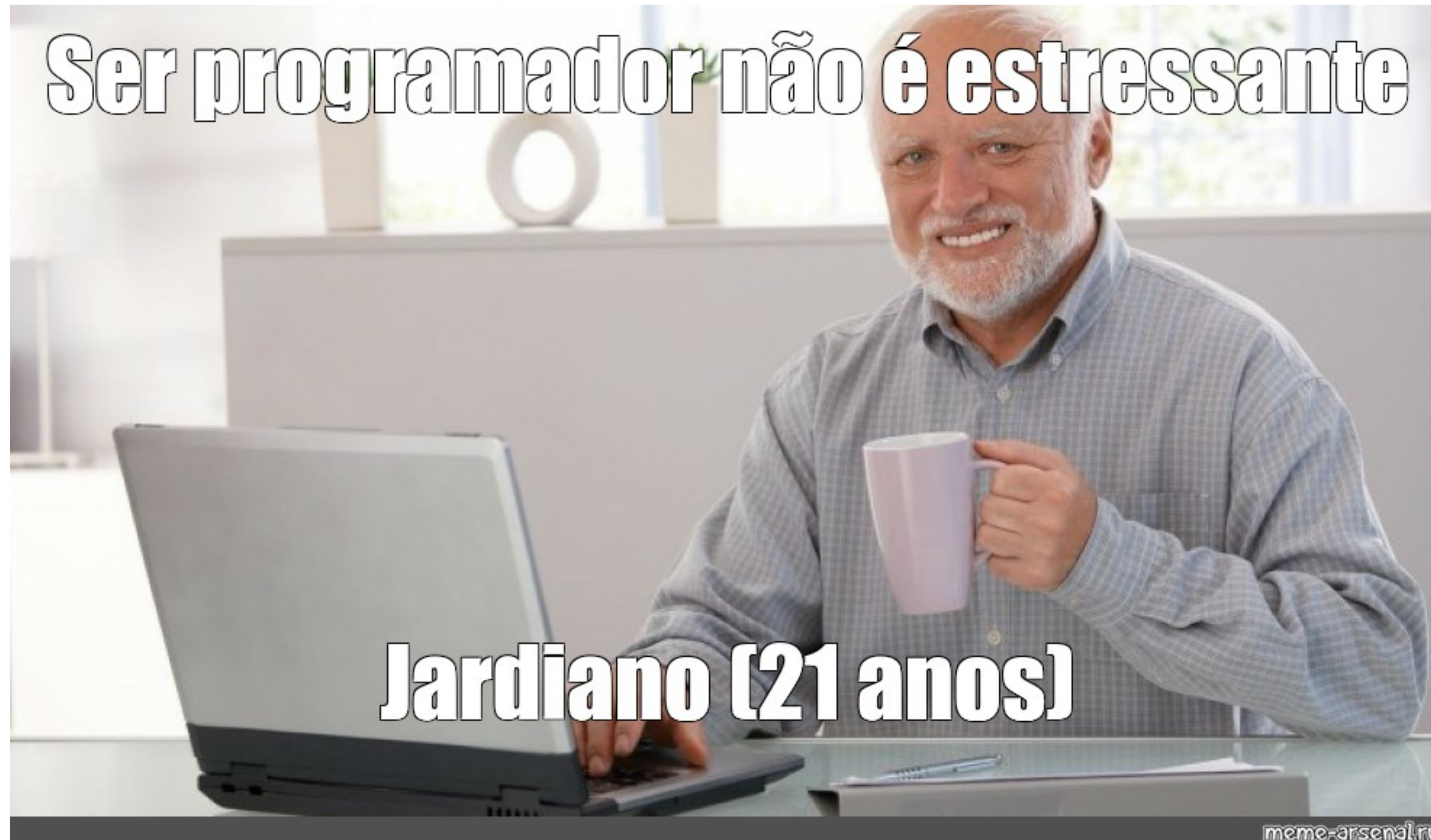
- É o ponto central com poderes de liderança sobre o produto: é o responsável por decidir quais recursos e funcionalidades serão construídos e em que ordem ficam no Product Backlog;
- Responsável por comunicar a todos os participantes uma visão clara do que a equipe Scrum está buscando alcançar no projeto: é o responsável pelo sucesso global da solução;
- Para garantir agilidade na construção do Produto ele deve colaborar ativamente com o Scrum Master e com os Developers, e deve estar disponível para responder às perguntas e dúvidas.

Papéis do Scrum: Scrum Master

Pode ser o representante do cliente no projeto e também desempenha um papel importante de facilitador. Trabalha junto com o Product Owner protegendo o Team de possíveis distúrbios externos, ou seja, o Scrum Master é um guardião do processo Scrum. É responsável por:

- Guiar e ajudar a treinar o Team para alcançar as metas dos lançamentos.
- Organiza a retrospectiva de Sprint para ajudar o Team a melhorar sua produtividade.
- Garante a colaboração entre as partes interessadas e a equipe e suas responsabilidades.
- Aplica os valores e as práticas Scrum.

Papéis do Scrum: Team (Devs)



Papéis do Scrum: Team (Devs, QA, UX, etc)

- São os membros da equipe que realmente executam o trabalho;
- Responsável por fazer as estimativas necessárias.
- Normalmente é uma equipe multidisciplinar, responsável por entregar o produto pronto e funcionando ao final de cada Sprint;
- A equipe de desenvolvimento se autogerência, determinando a melhor maneira de realizar o trabalho para atingir a meta estabelecida pelo Product Owner;
- A equipe deve ter coletivamente todas as habilidades necessárias para produzir, com qualidade, algo pronto e funcionando;

Cerimônias do Scrum

SPRINT PLANNING OU PLANEJAMENTO DA SPRINT

É a primeira reunião do projeto e todos precisam participar. Esta é a reunião em que o Product Owner planeja e elabora a lista de prioridades. Essa reunião deve ser dividida em duas partes:

Parte 1: o Product Owner define suas prioridades com base na história do usuário (requisitos), selecionando os itens do backlog que serão desenvolvidos e qual o objetivo da sprint. Esta primeira parte da reunião responde a questão “o quê” deve ser feito.

Parte 2: a equipe de desenvolvimento define a sprint backlog, que é um documento que contém as tarefas necessárias para cumprir a meta, deduz o quanto de tempo (em horas) será levada para concluí-las e transformá-las em incrementos de produto, que sejam potencialmente entregáveis.

Cerimônias do Scrum

DAILY MEETING OU DAILY OU REUNIÃO DIÁRIA

- Reunião diária entre os membros da equipe com tempo definido (cerca de 15 minutos);
- O objetivo é avaliar como o time está em relação à Meta da Sprint e produzir um plano de ação para o próximo dia de trabalho;
- Criar foco e melhorar o autogerenciamento.

É uma reunião em que cada participante deve responder a três perguntas:

1. O que você fez ontem?
2. O que você fará hoje?
3. Quais os obstáculos que impedem (ou podem impedir) seu trabalho?



Fonte: <<https://age-of-product.com/remote-daily-scrum-distributed-teams/>>

Cerimônias do Scrum

SPRINT REVIEW OU REVISÃO DA SPRINT

É uma reunião de balanço sobre tudo o que foi feito durante uma print. Nessa reunião o time deve mostrar os resultados da sprint para o Product Owner e seus convidados". Nessa reunião, são mostrados somente os itens concluídos, ou seja, caso tenha faltado uma atividade, o item não deve ser apresentado.

Participam da reunião: Product Owner, Scrum Master, Team e convidados. Ela tem duração estimada de quatro horas e é sempre marcada no final da sprint.

Durante a Revisão da Sprint, é avaliado o projeto com base nos objetivos da Sprint, que foram determinados durante o Planejamento da Sprint. É verificado se o Team completou cada um dos itens do Product Backlog e se o objetivo geral da Sprint foi atingido.

Cerimônias do Scrum

SPRINT RETROSPECTIVE OU RETROSPECTIVA DA SPRINT

A reunião Retrospectiva da Sprint tem por objetivo verificar o que foi bom e o que pode ser melhorado na sprint. Para Brod (2013), é uma reunião em que se "lava a roupa suja", ou seja, onde se procura entender o que deu errado e o que pode ser ajustado nas sprints para melhorar.

Participam dessa reunião: o Team e o Scrum Master, o Product Owner pode participar, caso seja convidado. Ela acontece após a reunião de revisão da Sprint e possui uma duração aproximada de três horas. Durante a reunião o Scrum Master deve tomar nota de tudo e o time deve priorizar os itens apontados em uma ordem de mudança. A retrospectiva é uma excelente forma de garantir a melhoria contínua do processo

Cerimônias do Scrum

SPRINT

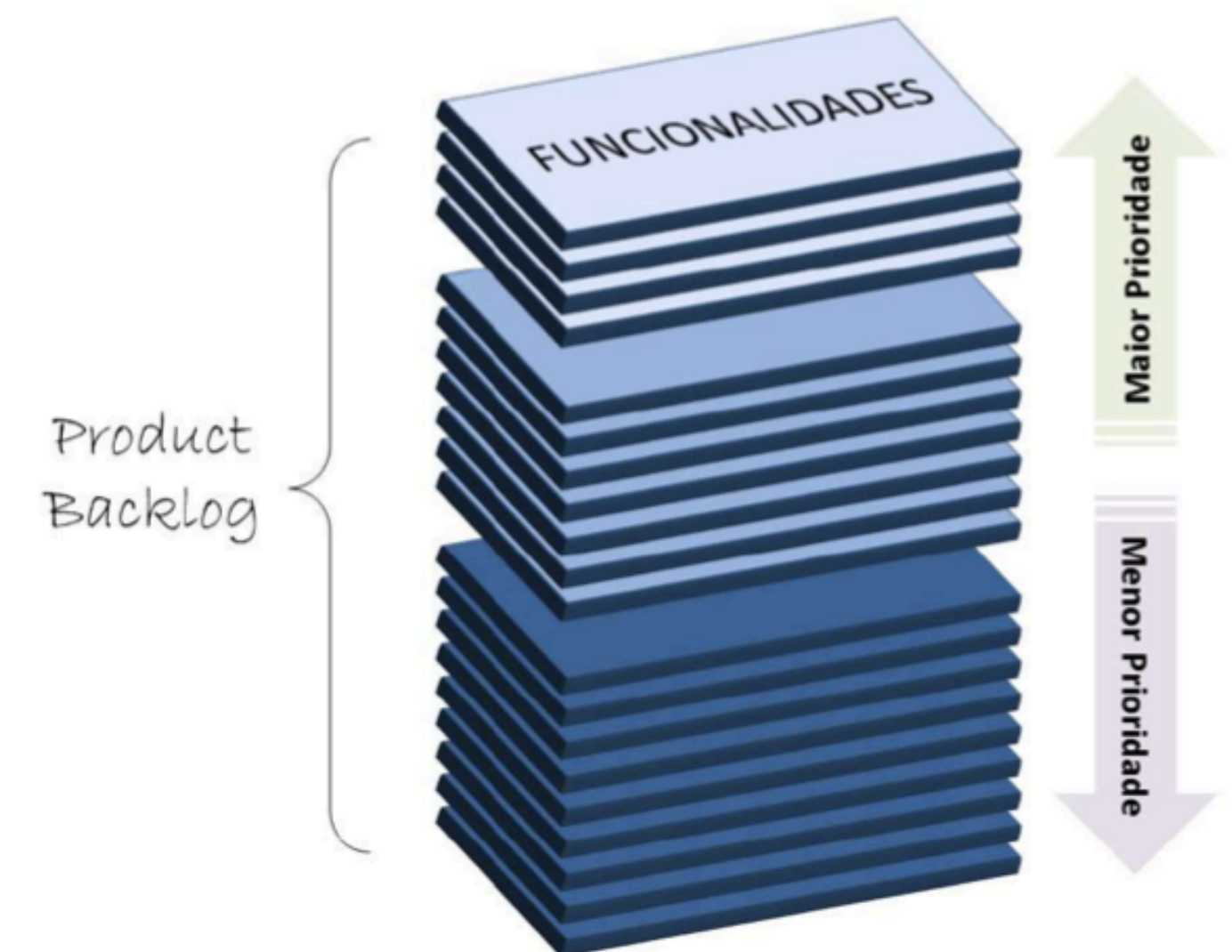
- O trabalho é realizado em iterações ou ciclos de até um mês, chamado de Sprints;
- O trabalho realizado em cada sprint deve criar algo de valor tangível para o cliente ou usuário;
- Sprints são timeboxed (duração fixa), com início e fim em data fixa, geralmente todos com a mesma duração;
- Um novo Sprint segue imediatamente a conclusão do Sprint anterior;
- Via de regra, não deve permitir nenhuma alteração de escopo ou pessoal durante um Sprint, embora esta regra possa ser quebrada devido à algumas necessidades de negócio.

Artefatos do Scrum

PRODUCT BACKLOG

O Product Backlog é um documento que segundo Sbrocco (2012, p. 168) “representa a visão do produto de forma modular, contendo todos os itens que devem ser desenvolvidos durante o projeto”. Ou seja, é uma lista de funcionalidades com prioridades que é feita no início do projeto com o objetivo de esclarecer o que deve ser entregue ao cliente.

Os itens do Product Backlog devem ser descritos de forma simples e de fácil entendimento para o cliente e para o Team e são criados e mantidos pelo Product Owner.



Artefatos do Scrum

SPRINT BACKLOG

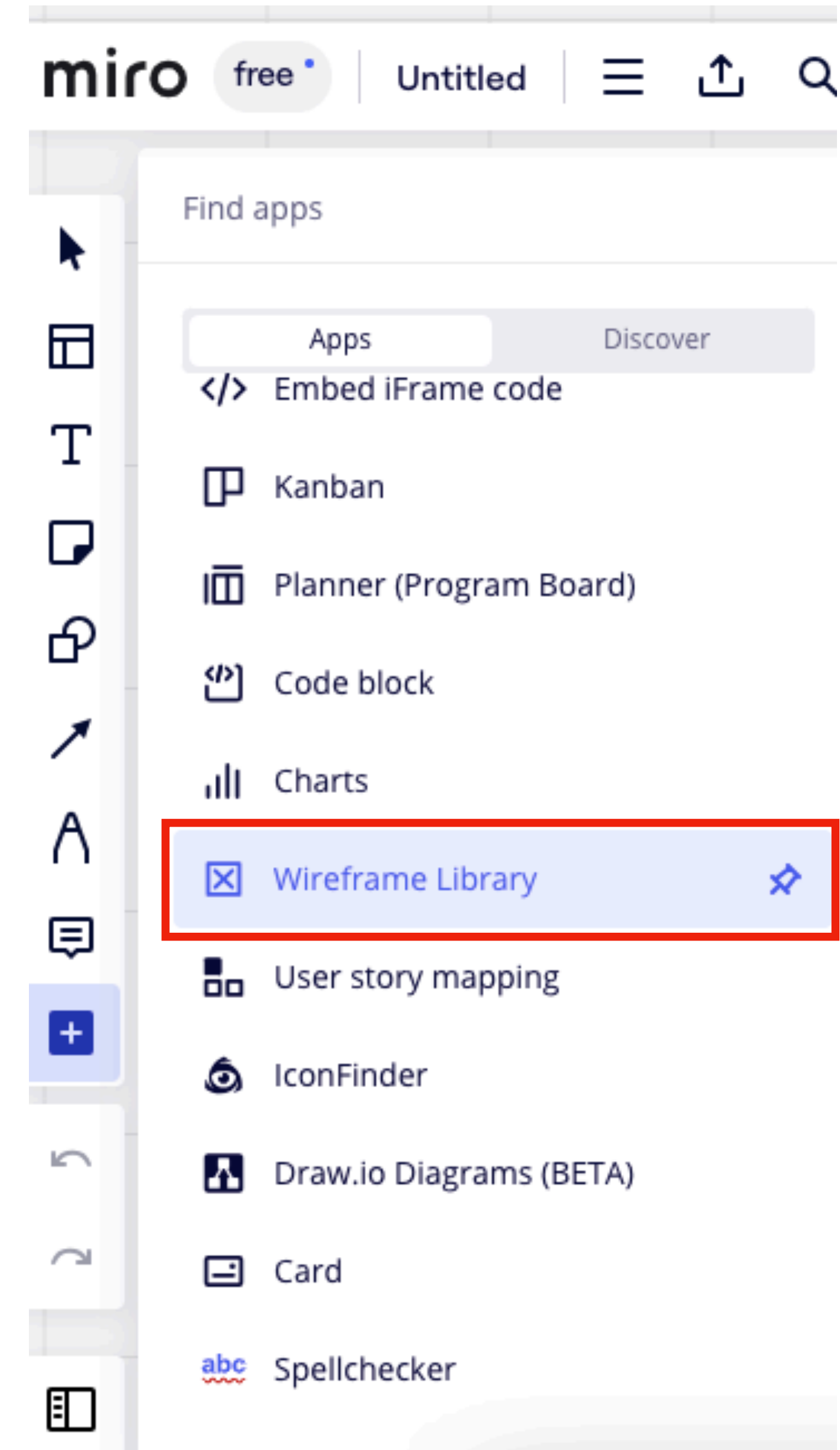
Artefato proveniente do Planejamento da Sprint e que conforme Sbrocco (2012, p. 168) “representa todas as tarefas que devem ser desenvolvidas durante uma Sprint ou iteração”. Para um melhor rendimento da Sprint, o Sprint Backlog não pode sofrer alterações e é importante que o Team conclua o primeiro item, antes de começar o segundo, para evitar riscos na meta.

Pode ser subdividida as tarefas entre equipes distintas e, com isso, ter várias sprints ocorrendo simultaneamente.

Desenvolvendo protótipo

Desafio:

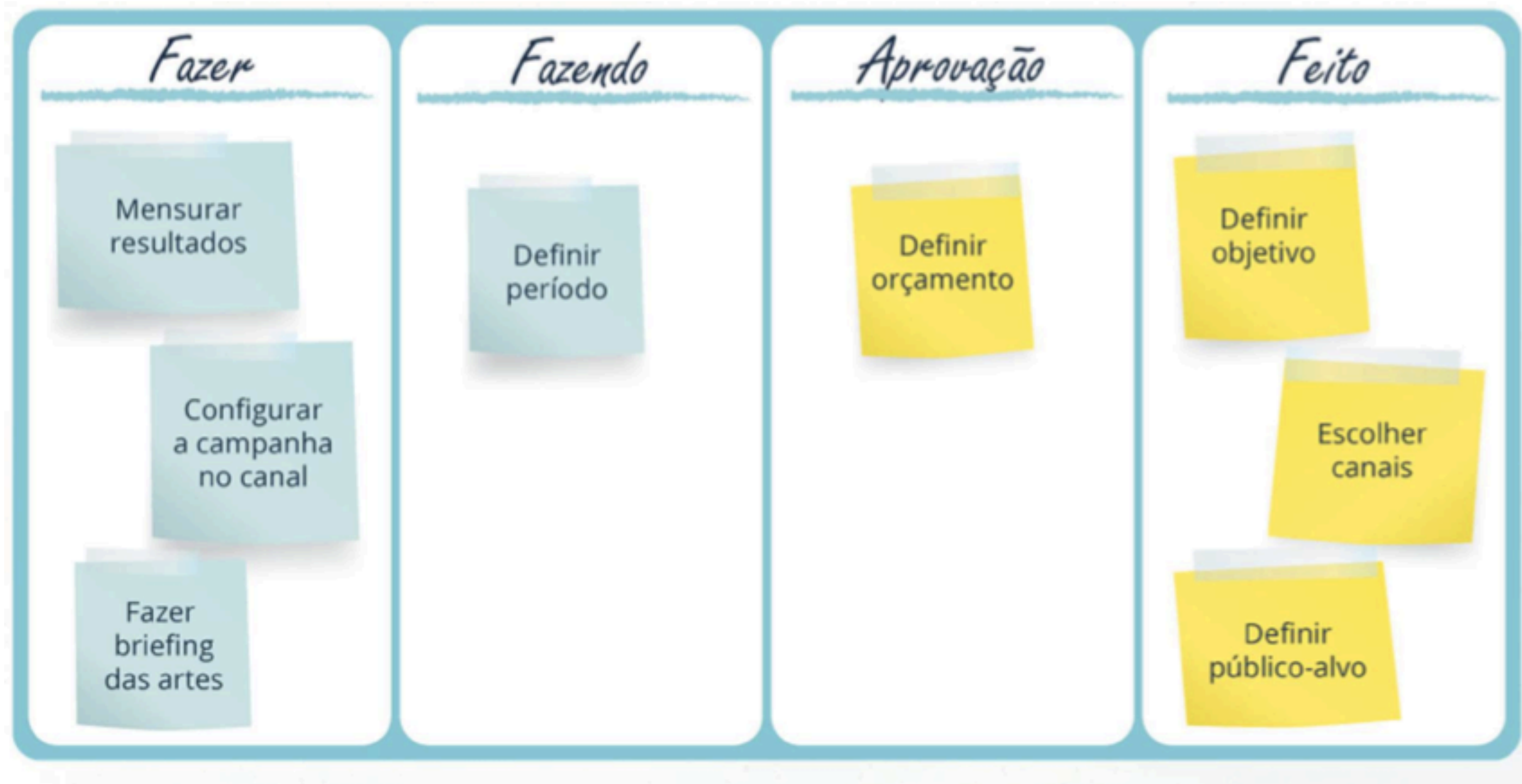
<https://miro.com/>



Método Kanban

- A palavra japonesa *kanban* significa “sinalização” ou “cartão” e poderia ser usada para fazer referência a um **sistema de sinalização que utiliza cartões**.
- Diversas metodologias e *frameworks* utilizados na agilidade, como o Scrum, por exemplo, utilizam-se de um quadro que podemos chamar “Quadro Kanban”, no qual temos alguns cartões que se deslocam pelo quadro, ou seja, alguns “kanbans”. Por isso, é fundamental ter em mente o que é Scrum e Kanban, para não se confundir.
- Kanban valoriza o fluxo das atividades, entrega contínua, visualização do trabalho, eliminação do desperdício e melhoria contínua.

Método Kanban



Método Kanban

- Trabalha com a visão de trabalho puxado, e não empurrado (como é no Scrum);
- Os cartões que já começaram a ser trabalhados são considerados como WIP, que significa *work in progress*;
- À medida que um item é finalizado, o WIP diminui e um espaço de trabalho é liberado;
- Com isso, uma nova tarefa poderá ser feita, caracterizando um sistema puxado de trabalho;
- Limitar o trabalho em progresso (WIP) é importante para que não haja excesso de trabalho dentro do fluxo: ao limitar o WIP, a entrada de novas tarefas é baseada na disponibilidade de espaços de trabalho no fluxo;
- No Kanban, não há timebox para entrega de tarefas (sprint), ela é contínua;
- Se o cartão chegou na etapa final, o cliente já pode utilizá-lo;
- No Scrum, é necessário esperar o final do sprint.

Método Kanban

Cartão:

O cartão é a representação de uma tarefa ou ação que precisa ser tomada para que o resultado final seja entregue.

Geralmente eles são diferenciados por um sistema de cores, que categoriza a ação e indica: quem é o responsável pela tarefa; qual o nível de prioridade; ou qual o tipo da tarefa.

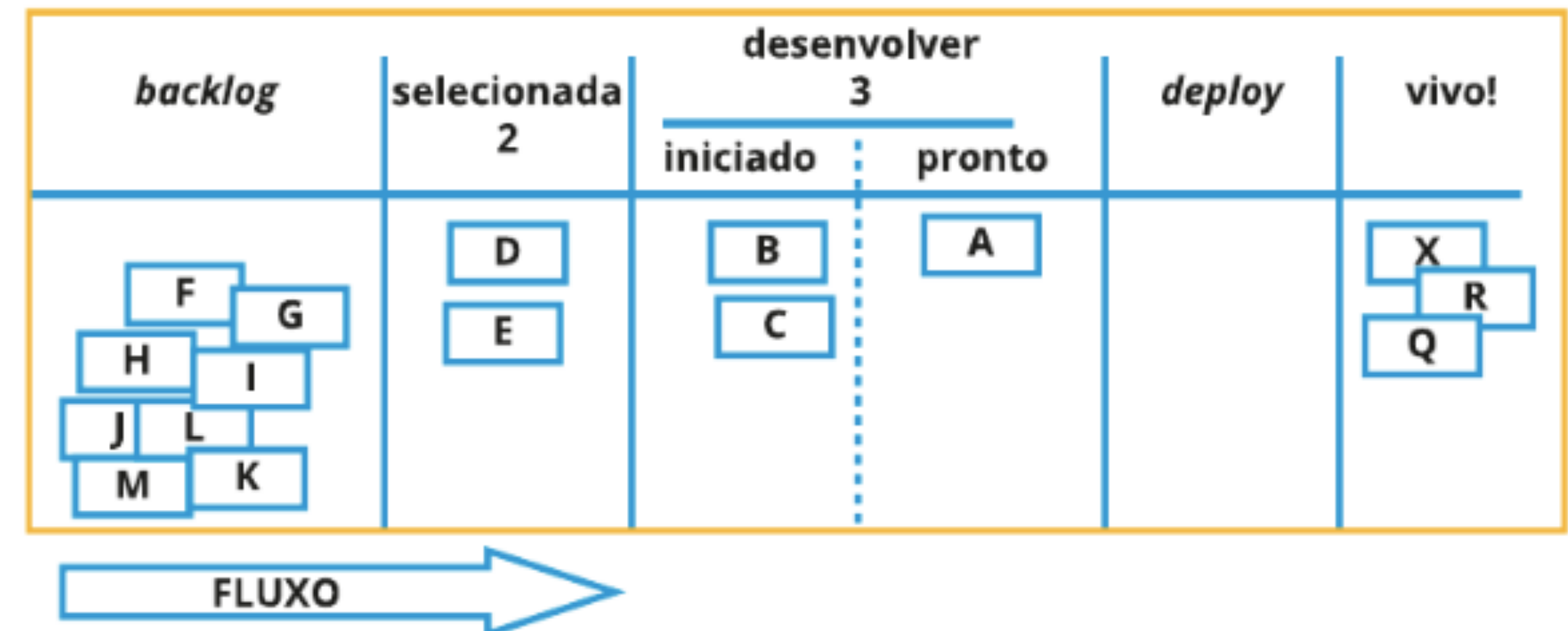
Em um projeto de construção de um site, um cartão poderia ser “produzir imagem para a página inicial”, por exemplo.



Método Kanban

Colunas

As colunas representam os status da atividade. Um kanban geralmente possui três colunas: "a fazer", "em execução" e "feito", mas essas colunas podem ser nomeadas de acordo com as necessidades da organização.



Quadro

O quadro nada mais é do que a junção visual das colunas e cartões, representando o sistema kanban como um todo.

Cada quadro é um kanban e uma única equipe pode trabalhar com vários quadros simultaneamente.

Método Kanban

Como funciona?

A utilização do kanban consiste basicamente em mover os cartões conforme o status for alterado, dando um panorama do que está pendente e do que já foi concluído;

Além disso, outro ponto importante é o caráter limitante do sistema kanban, porque uma linha produtiva pode ter uma demanda muito maior do que a capacidade de produzir;

No kanban, só serão adicionadas as atividades que o time comporta, sendo as novas demandas inseridas conforme as entregas forem realizadas.

XP - eXtreme Programming

O XP - eXtreme Programming ou Programação Extrema é uma metodologia rigorosa e disciplinada. É uma filosofia de desenvolvimento de software fundamentada em quatro valores

- Comunicação
- Feedback
- Simplicidade
- Coragem

Enquanto o Scrum tem maior foco nas práticas ligadas ao gerenciamento e organização e Kanban é focado no fluxo, o XP dá mais atenção às práticas de programação.

XP - eXtreme Programming

Comunicação: Valoriza a comunicação aberta e constante entre todos os membros da equipe de desenvolvimento, incluindo programadores, clientes e gerentes. A comunicação é vista como uma chave para entender requisitos, resolver problemas e manter todos alinhados com os objetivos do projeto.

Feedback: Prioriza o feedback rápido e contínuo. Isso envolve a realização de revisões de código, testes automatizados e a obtenção de feedback dos clientes para garantir que o software atenda às expectativas e seja de alta qualidade.

Simplicidade: Valoriza a simplicidade na concepção e implementação do software. A ideia é manter o código simples e direto, evitando a complexidade desnecessária que pode dificultar a manutenção e a compreensão do sistema.

Coragem: Encoraja a coragem da equipe para tomar decisões ousadas e enfrentar desafios difíceis. Isso pode incluir a vontade de refatorar o código, fazer mudanças significativas e buscar melhorias contínuas, mesmo que isso signifique sair da zona de conforto.

TDD- Test Driven Development

TDD- Test Driven Development ou Desenvolvimento Dirigido por Testes, você testa o código antes de colocá-lo em execução. O objetivo principal é a especificação, não a validação.

Muitos o consideram uma técnica de desenvolvimento. Alguns também o consideram como uma forma de criar um código limpo. Mas o fato preponderante é que atua como um grande apoio na qualidade do desenvolvimento de software.

Ferramentas que auxiliam no desenvolvimento:

- .NET - NUnit
- PHP - PHPUnit
- Node ou Javascript - Jasmine
- Java - JUnit
- Python - PyUnit

Bibliografia:

PAULA FILHO, Wilson de Pádua. Engenharia de Software: Fundamentos, Métodos e Padrões. LTC, 2009.

PRESSMAN, R. S. Engenharia de software. 6.ed. McGraw-Hill, 2006.

SOMMERVILLE, I. Engenharia de software. 8.ed. Addison Wesley, 2007.

SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar de. Metodologias ágeis : engenharia de software sob medida . São Paulo: Érica, 2012.