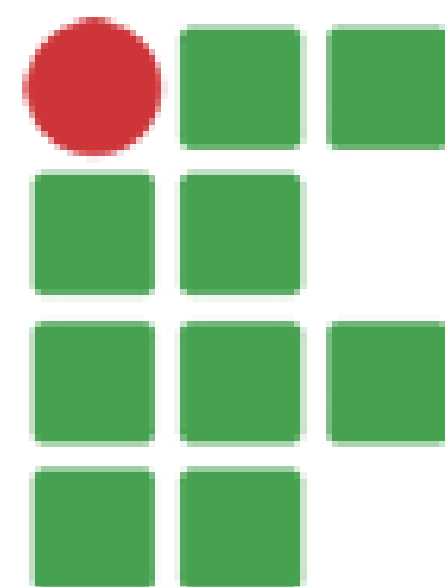




**INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
Rio de Janeiro

Disciplina

Modelagem de Domínio e
Conceitos de Orientação a Objetos

PROF. FERNANDO TAMBERLINI ALVES

- **Engenharia de Software:** processos, métodos e ferramentas de apoio que nos auxiliam no desenvolvimento de software;
- **Arquitetura de Software:** características de como o software será desenvolvido;
- **Processo de Desenvolvimento:** etapas específicas a serem percorridas durante o desenvolvimento de um software (Cascata e Ágil);
- **Levantamento de Requisitos:** é o processo de entender o que o sistema deve fazer (requisitos funcionais) e como ele deve se comportar (requisitos não funcionais).
- **Modelagem:** é a representação graficamente a estrutura, o comportamento e os requisitos do sistema antes de sua construção.

Engenharia de Software

- **Processo:** refere-se ao conjunto de ações sequenciais para atingir um resultado. Envolve várias etapas interdependentes que precisam ser seguidas para chegar a um objetivo final. É mais amplo que o método, pois abrange o contexto, as etapas e a organização das atividades.
- **Método:** refere-se ao como algo é feito. É uma abordagem estruturada para realizar uma tarefa ou alcançar um objetivo. Envolve técnicas, passos ou práticas específicas, definindo a forma de execução
- **Ferramenta:** é um programa ou aplicação projetada para auxiliar a equipe responsável no desenvolvimento, manutenção e gerenciamento do software;



- **Ferramentas de Desenvolvimento IDE** (Ambiente de Desenvolvimento Integrado) como Visual Studio que oferecem editores de código, depuradores e outras funcionalidades integradas.
- **Compiladores e Interpretadores** que traduzem o código-fonte para um formato executável pelo computador.
- **Controle de Versão** como as ferramentas Git, SVN e Mercurial ajudam a gerenciar e acompanhar as mudanças no código ao longo do tempo, facilitando a colaboração e a reversão de alterações.
- **Ferramentas de Modelagem e Design**, como Lucidchart e Draw.io, que permitem a criação de diagramas e modelos que representam a arquitetura e o design do sistema.

- Civil, mecânica, elétrica, aviação, automobilística, etc
- Projeto com duas características:
 - Planejamento detalhado (big upfront design)
 - Sequencial
- Processo estabelecido e com poucas mudanças
 - Similar ao Processo Cascata (Waterfall)

Software é diferente !

- Engenharia de Software $\neq$ Engenharia Tradicional
- Software $\neq$ (carro, ponte, casa, avião, celular, etc)
- Software $\neq$ (produtos físicos)
- Software é abstrato e "adaptável"

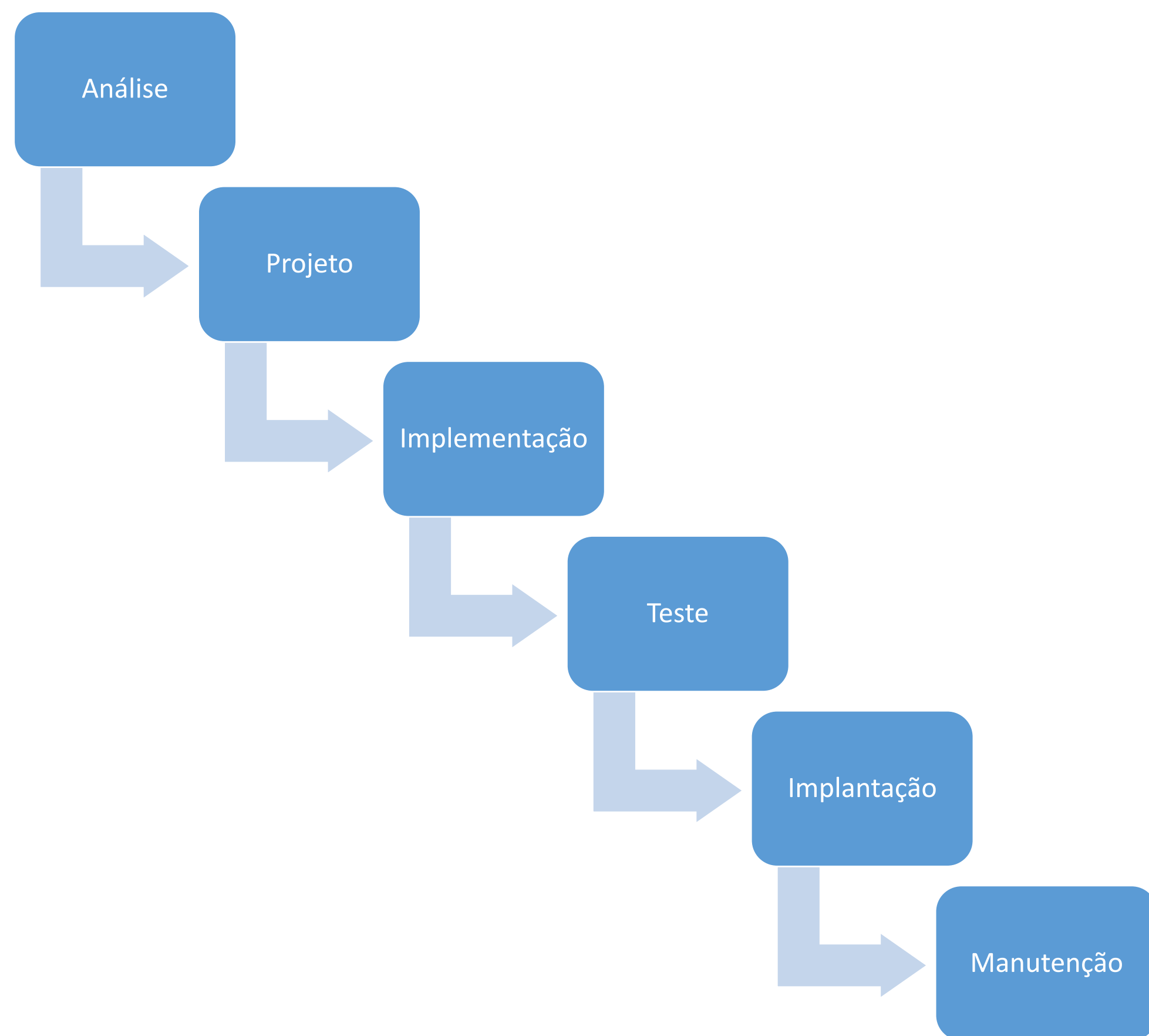
1ª Dificuldade – Requisitos

- Clientes não sabem o que querem (em um software)
- Funcionalidades são "infinitas" (difícil prever)
- Mundo muda!
- Não dá mais para ficar 1 ano levantando requisitos, 1 ano projetando, 1 ano implementando, etc
- Quando o software ficar pronto, ele estará obsoleto!

2ª Dificuldade – Detalhamento Excessivo

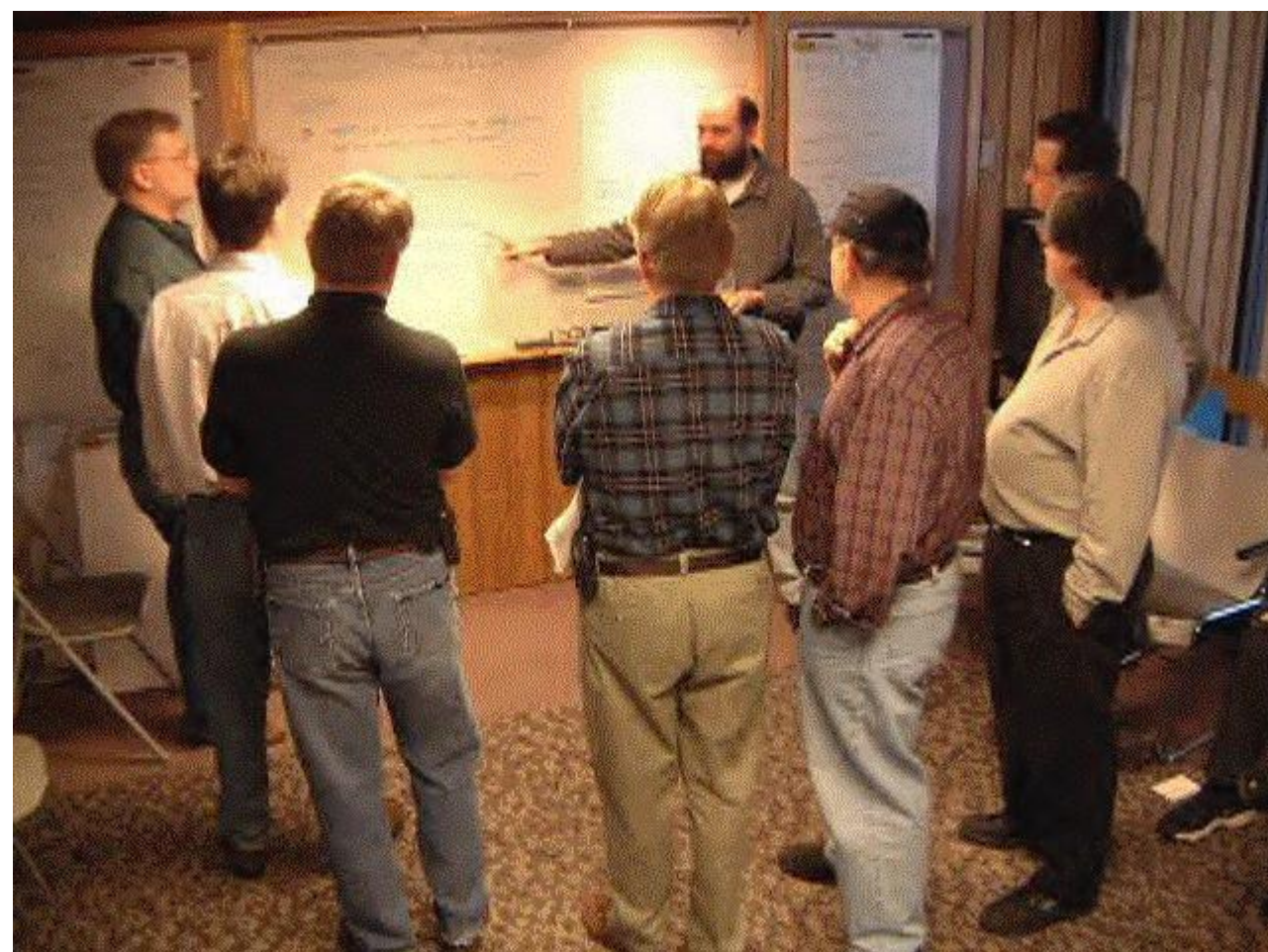
- Verbosas e pouco úteis;
- Na prática desconsideradas durante a implementação;
- *Plan-and-document* não funcionou com software

Processo - Cascata



- **Análise (planejamento):** definição dos objetivos, escopo e recursos necessários. Identificação e das funcionalidades e restrições do software;
- **Projeto (design) :** definição da arquitetura adequada e design do sistema
- **Implementação (codificação):** codificação e construção do software conforme projeto;
- **Teste:** verificação e validação do software para garantir qualidade e correção.
- **Implantação:** Lançamento do software no ambiente de produção.
- **Manutenção:** Atualização, correção de erros e melhorias contínuas após a implantação.

Manifesto Ágil



Indivíduos e interações
mais do que processos e ferramentas

01

Softwares em funcionamento
mais do que documentação abrangente

02

Responder as mudanças
mais do que seguir um plano

03

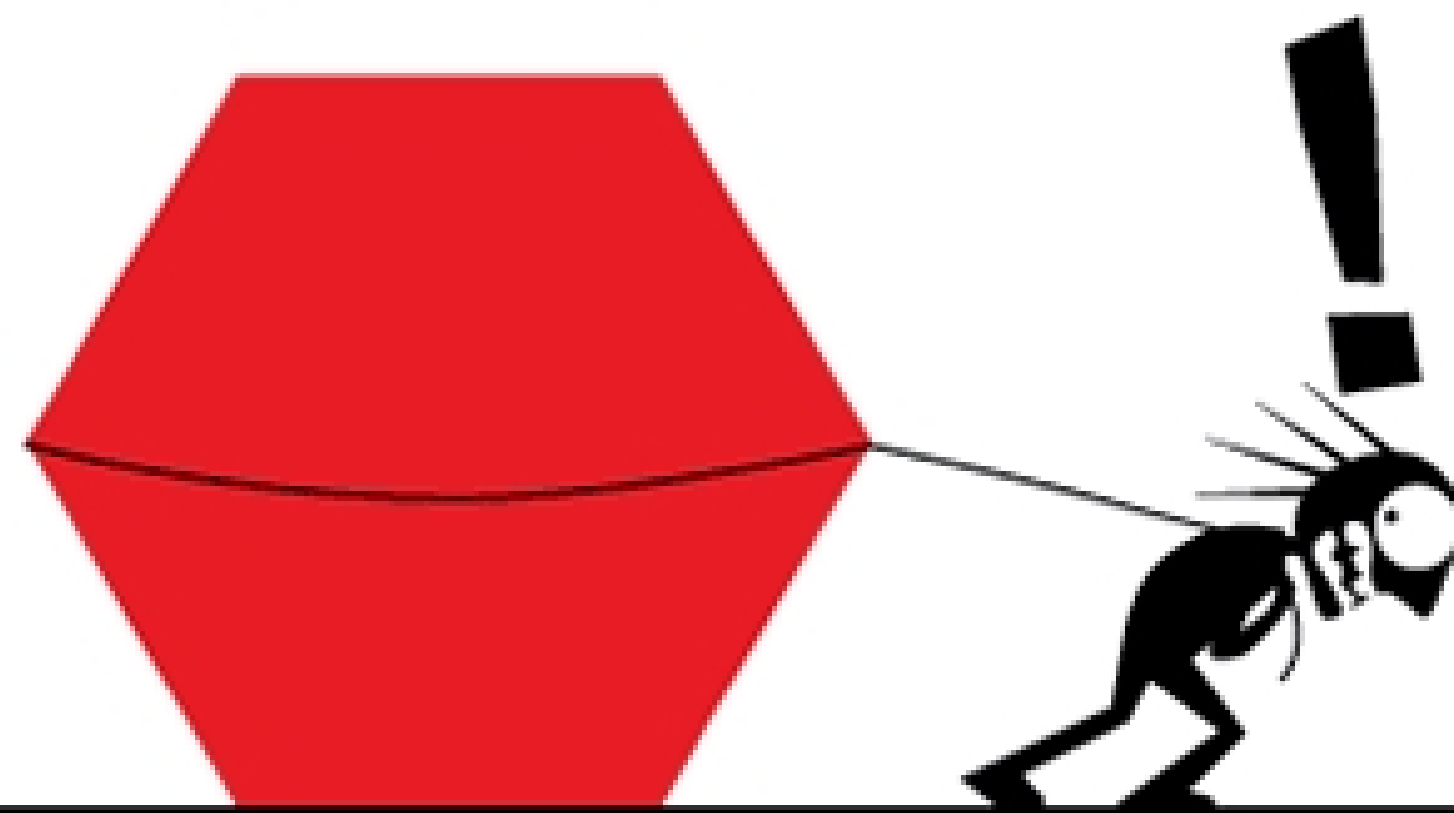
Colaboração com o cliente
mais do que negociação de contratos

04

Processo - Ágil

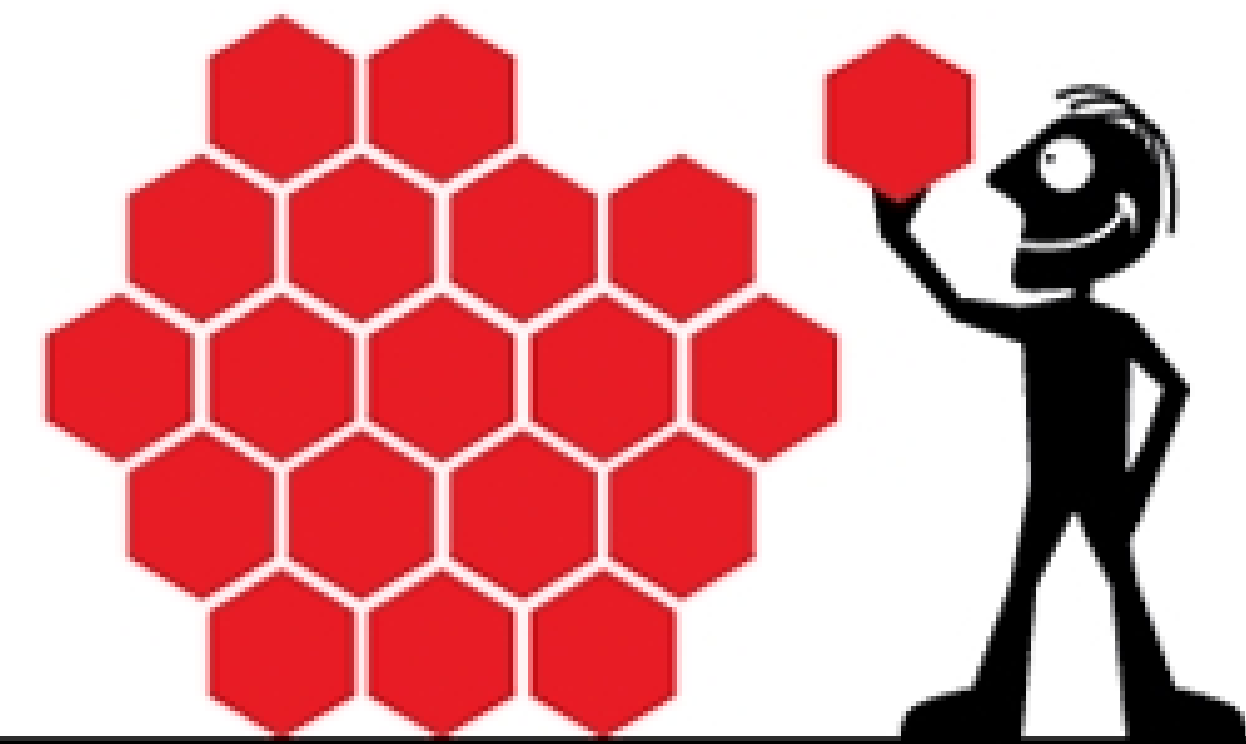
Ágil = iterativo

THE WATERFALL PROCESS



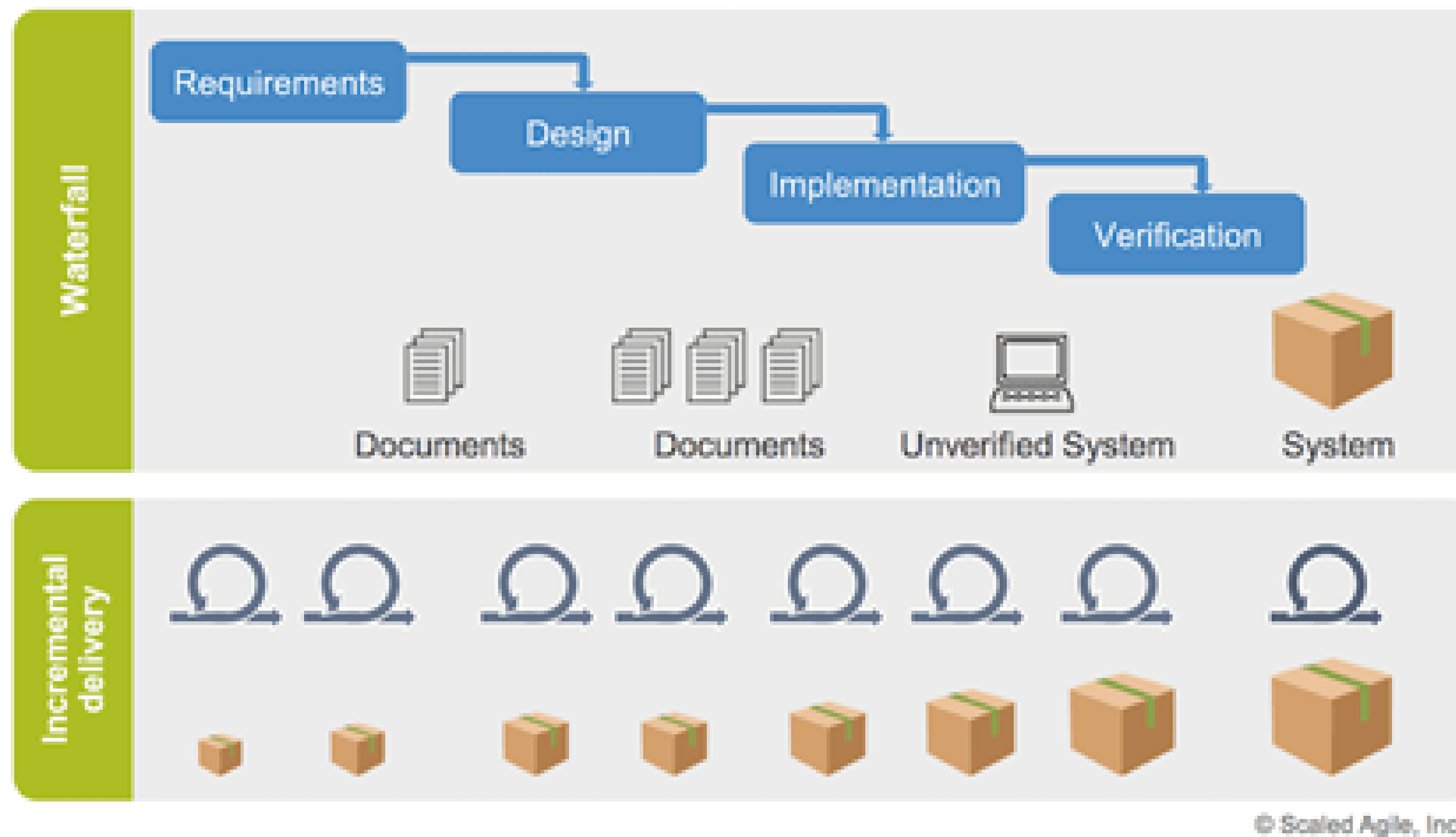
Este projeto é muito grande!
E se precisarmos mudar algumas necessidades?

THE AGILE PROCESS

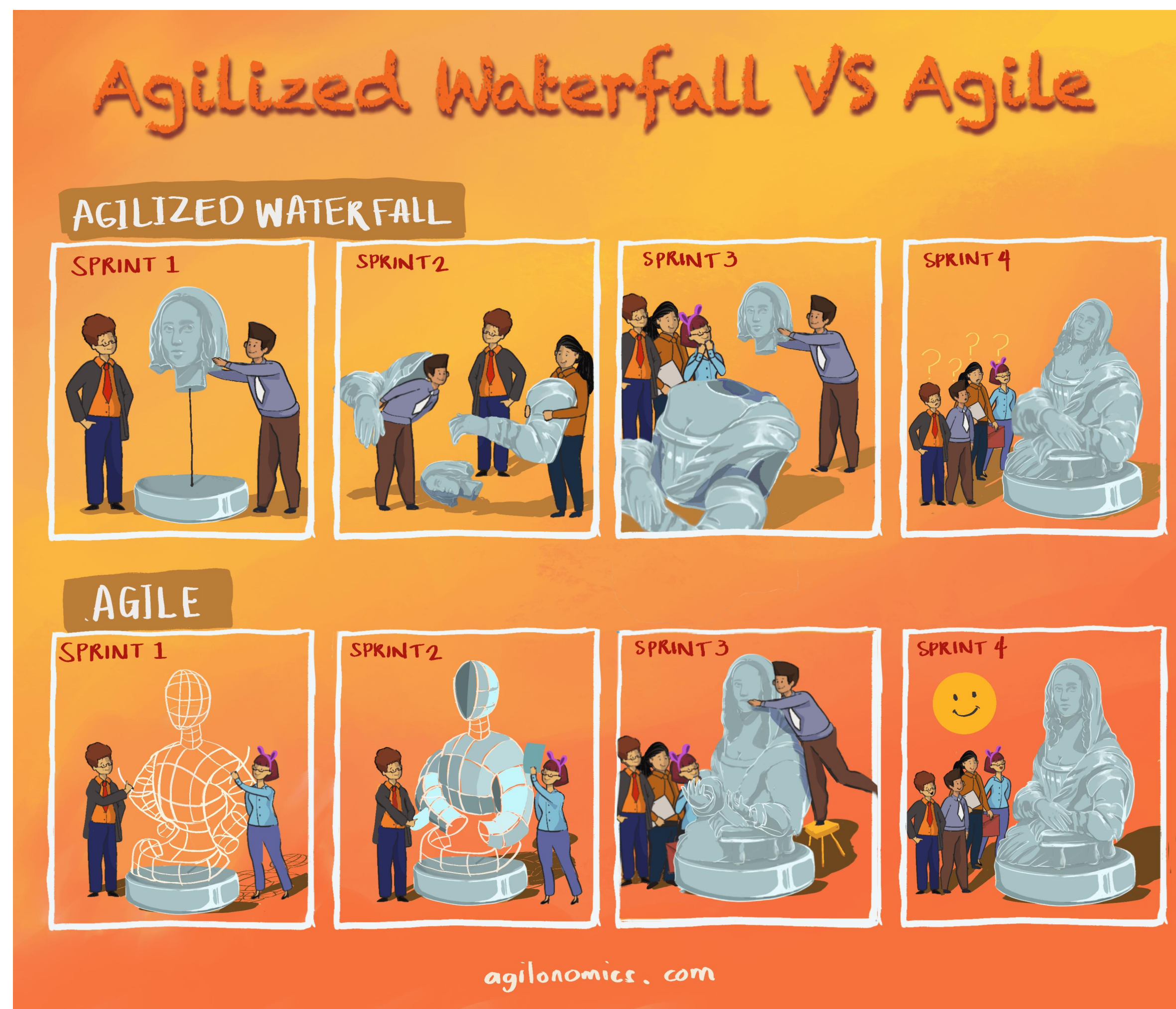


Vamos dividir o projeto em pequenas partes!
Podemos realizar mudanças a qualquer tempo.

Processo - Ágil



Processo – Agilezed Waterfall



- Extreme Programming (XP)
- Scrum
- Kanban

Valores + Princípios + Práticas

Valores:

- Comunicação, Simplicidade, Feedback, Coragem, Respeito e Qualidade do Ambiente de Trabalho

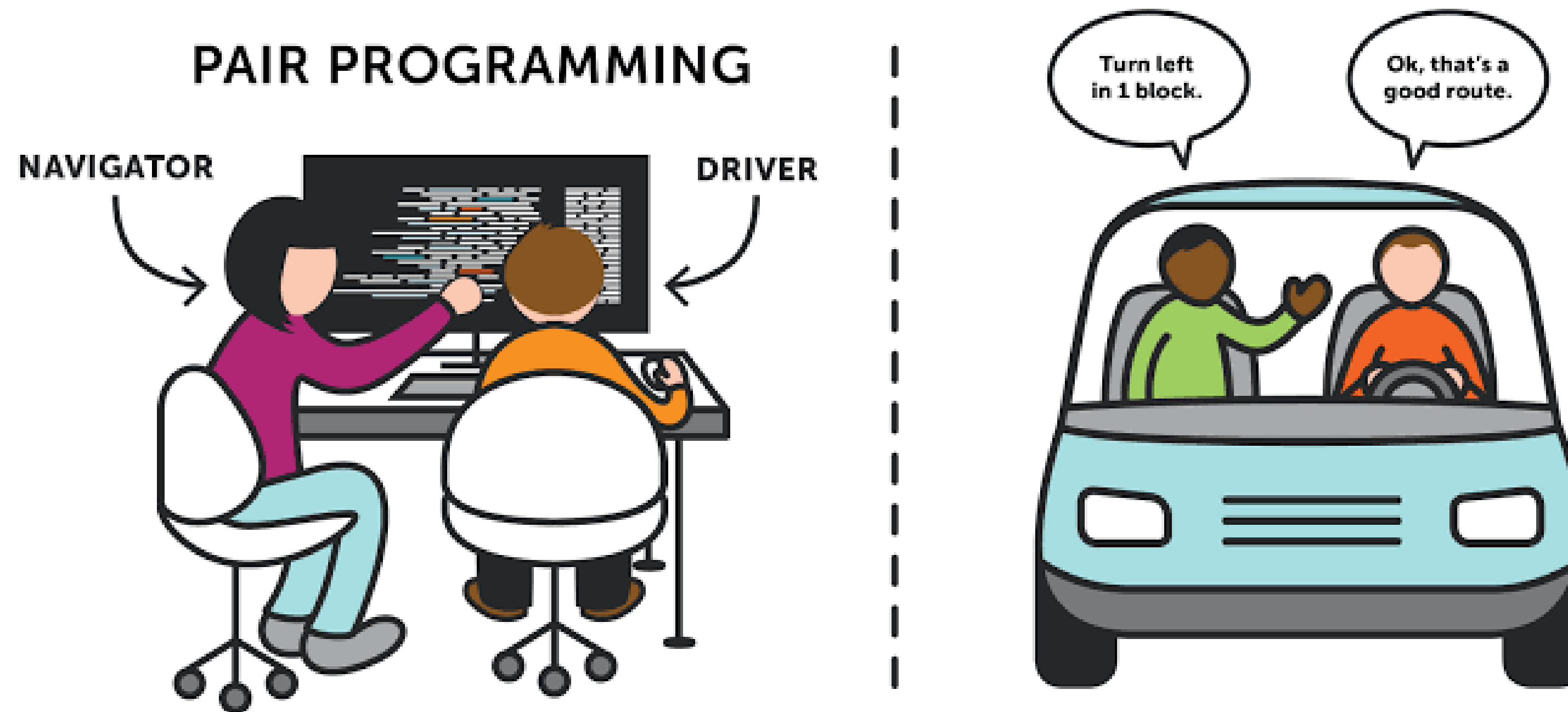
Princípios:

- Viabilidade econômica, Melhorias Contínuas, Falhas Acontecem, Baby Steps e Responsabilidade

Práticas:

- Histórias de Usuário, Programação Pareada, Desenvolvimento Dirigido por Testes etc...

Pair Programming



Métodos Ágeis - Scrum

Framework ágil para gestão e desenvolvimento de projetos complexos.

Papéis Principais:

Product Owner – define o que deve ser feito

Scrum Master – garante que o time siga o Scrum

Time de Desenvolvimento – entrega incrementos do produto

Eventos:

Sprint – ciclo de trabalho (1-4 semanas)

Daily Scrum – reunião diária de 15 min

Sprint Planning – planejamento da Sprint

Sprint Review – demonstração do que foi feito

Sprint Retrospective – melhoria contínua

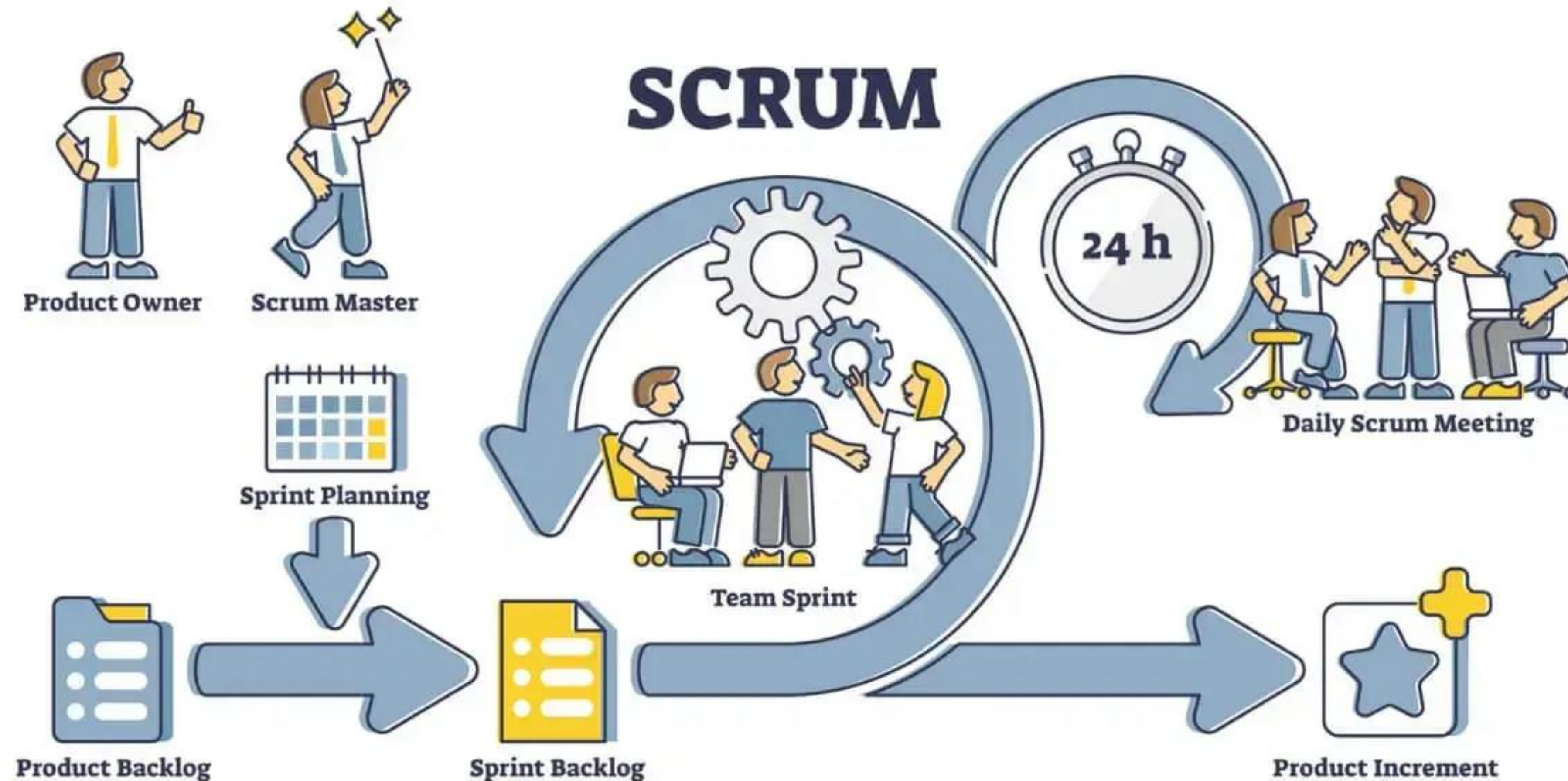
Artefatos:

Product Backlog – lista priorizada de requisitos

Sprint Backlog – tarefas da Sprint

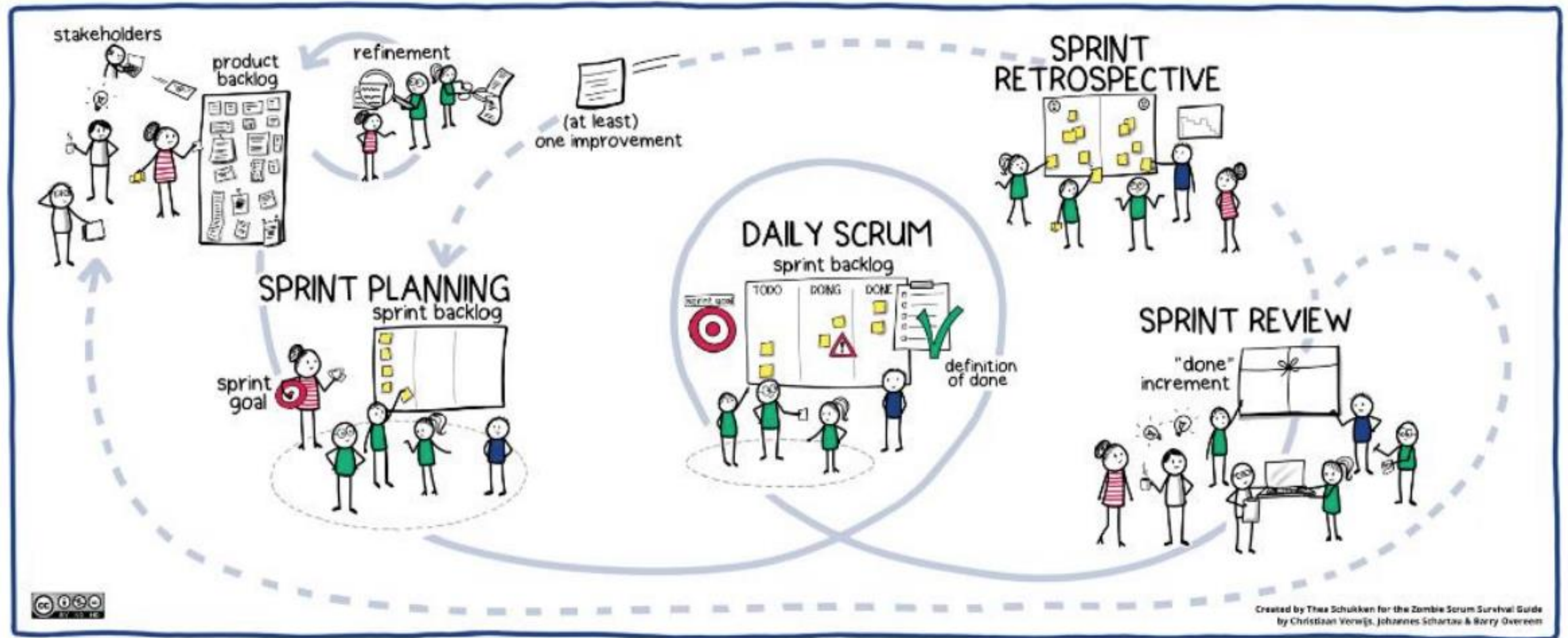
Incremento – entrega funcional ao final da Sprint

Métodos Ágeis - Scrum



Fonte: <https://blogdaqualidade.com.br/scrum-o-que-e-e-quando-usar/>

Métodos Ágeis - Scrum



fonte: <https://www.scrum.org/resources/scrum-framework-reduce-risk-and-deliver-value-sooner>

Métodos Ágeis - Kanban

Método visual para gerenciamento de fluxo de trabalho e melhoria contínua.

Princípios:

- Comece com o que você faz hoje
- Concorde em buscar mudanças incrementais
- Respeite processos, papéis e responsabilidades atuais
- Incentive a liderança em todos os níveis

Práticas Fundamentais:

- Visualizar o fluxo de trabalho (Quadro Kanban)
- Limitar o trabalho em progresso (WIP)
- Gerenciar o fluxo
- Ter políticas explícitas
- Usar feedbacks frequentes
- Melhorar colaborativamente

Kanban Típico

:  To Do →  Doing →  Done

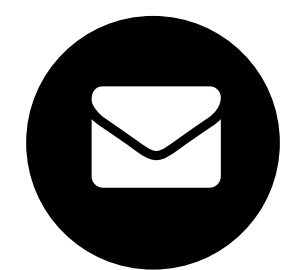
Métodos Ágeis - Kanban



Dúvidas?



Dúvidas?



fernando.alves@ifrj.edu.br



ftamberlini.dev.br