De: O Mentor

*Bem-vindo à **Parte III: Gestão Ágil**. Planejar continua importante, mas projetos de software aprendem durante a execução. Neste capítulo, estudaremos os valores do **Manifesto Ágil** e o **Scrum** como um framework para inspecionar resultados e adaptar o trabalho com responsabilidade compartilhada.*

Acesse a versão completa em: https://alessiojr.com/disciplines/gps/conteudo/00-livro/

Versão compilada em 20/9/2026 – Capítulo isolado do livro.

1 Manifesto Ágil e planejamento adaptativo

A Figure 1 situa o contexto de adaptação discutido nesta seção. O desenvolvimento de software frequentemente envolve incerteza: requisitos podem ser esclarecidos, hipóteses técnicas podem falhar e pessoas usuárias podem fornecer novas informações. Em 2001, os signatários do **Manifesto Ágil** formularam quatro valores para orientar esse contexto:

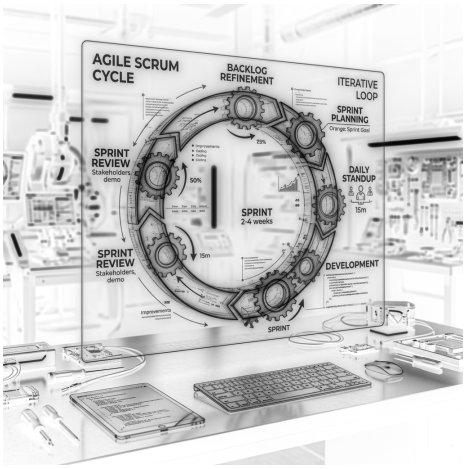


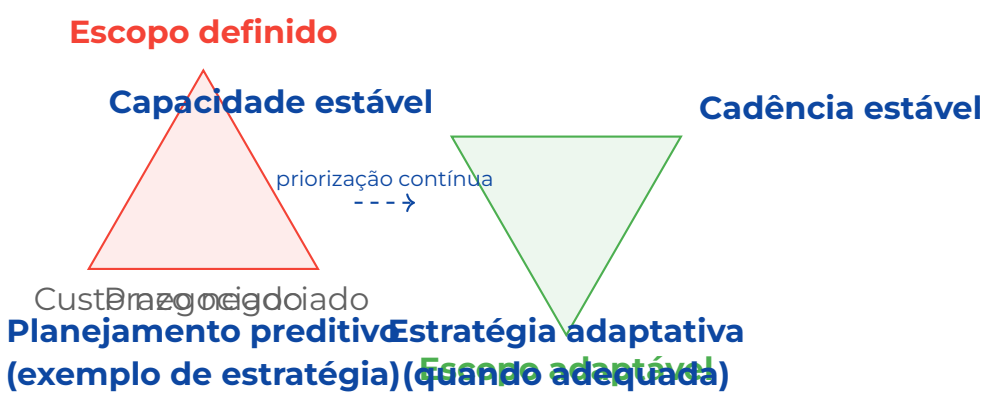
Figure 1: Ilustração de abertura: adaptação e colaboração no desenvolvimento ágil. *Fonte: Imagem gerada por IA, com supervisão do autor.*

1. **Indivíduos e interações** mais que processos e ferramentas.
2. **Software em funcionamento** mais que documentação abrangente.
3. **Colaboração com o cliente** mais que negociação de contratos.
4. **Responder a mudanças** mais que seguir um plano.

Os itens à direita continuam tendo valor. A ideia não é abandonar planejamento, documentação ou contratos, mas usar evidências do produto e do contexto para revisá-los quando necessário. Protótipos, testes e demonstrações podem reduzir incertezas antes de investimentos maiores.

1.1 Restrições e escopo

Todo projeto administra relações entre escopo, prazo, custo, qualidade e risco. Manter prazo e capacidade da equipe relativamente estáveis, ajustando o escopo conforme o aprendizado, é uma estratégia comum em produtos digitais. Ela pode facilitar previsibilidade de investimento e priorização de valor, mas não é uma regra do Scrum nem se aplica automaticamente a qualquer contrato ou organização.



Acesse a versão completa em: https://alessiojr.com/disciplines/gps/conteudo/00-livro/

Versão compilada em 20/9/2026 – Capítulo isolado do livro.

Acesse a versão completa em: https://alessiojr.com/disciplines/gps/conteudo/00-livro/

Figure 2: Estratégias ilustrativas para administrar restrições de projeto. *Fonte: Elaboração do autor.*

A Figura 2 sintetiza visualmente o conceito discutido nesta seção.

2 Scrum e empirismo

O **Scrum** é um framework leve para gerar valor por meio de soluções adaptativas para problemas complexos. Segundo o *Scrum Guide 2020*, ele se baseia em empirismo e pensamento enxuto. Transparência torna visível o estado relevante do trabalho; inspeção avalia artefatos e progresso; adaptação ajusta produto ou modo de trabalho quando necessário. Transparência não exige um quadro Kanban: este é uma técnica visual que uma equipe pode adotar.

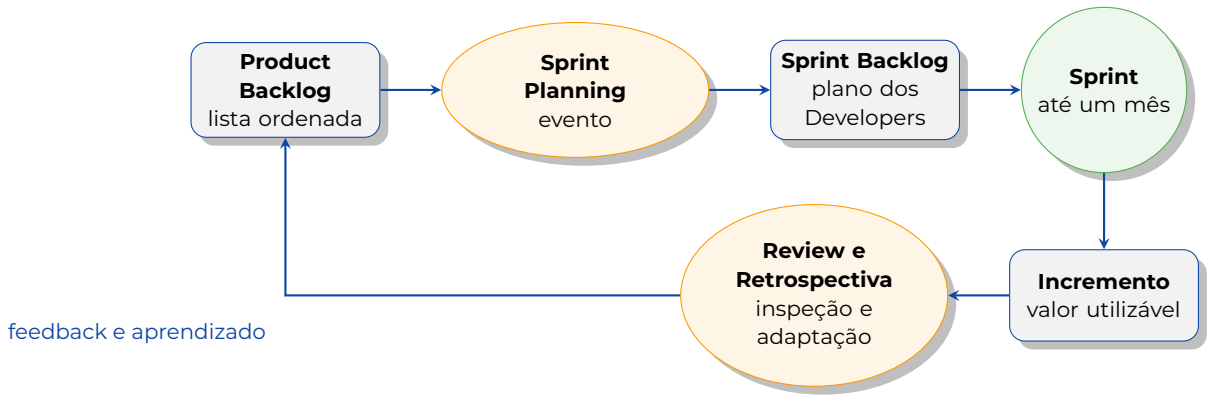


Figure 3: Fluxo didático entre artefatos, eventos e Incremento no Scrum. *Fonte: Elaboração do autor.*

A Figura 3 sintetiza visualmente o conceito discutido nesta seção.

2.1 Scrum Team

O Scrum Team é uma pequena equipe sem subequipes ou hierarquias internas, composta por um **Product Owner**, um **Scrum Master** e **Developers**. Seus membros colaboram para o Product Goal.

- **Product Owner (PO):** maximiza o valor do produto e responde pela gestão eficaz do Product Backlog, inclusive por sua ordenação e clareza. Pode delegar tarefas, mas permanece responsável por esse resultado.
- **Scrum Master (SM):** é responsável por estabelecer o Scrum e por sua efetividade. Serve à equipe, ao PO e à organização, facilitando a remoção de impedimentos e promovendo práticas coerentes com o framework. Não é gerente técnico nem precisa resolver pessoalmente um incidente de infraestrutura: pode apoiar a equipe a encontrar quem o resolva.
- **Developers**: são as pessoas comprometidas em criar qualquer aspecto de um Incremento utilizável. Eles definem como transformar itens selecionados em valor e adaptam seu plano diariamente.

Versão compilada em 20/9/2026 – Capítulo isolado do livro.

Acesse a versão completa em: https://alessiojr.com/disciplines/gps/conteudo/00-livro/

2.2 Sprint, eventos e artefatos

Uma **Sprint** é um evento de duração fixa de **até um mês**; duas semanas são a cadência adotada pela disciplina quando aplicável, não uma exigência universal. Sprints ocorrem em sequência e contêm todos os demais eventos necessários para criar um Incremento. Sua duração é escolhida para equilibrar feedback, risco e custo de planejamento.

Na **Sprint Planning**, todo o Scrum Team define por que a Sprint é valiosa, o que pode ser feito e como o trabalho será realizado. O **Sprint Goal** fornece o objetivo único da Sprint. Os Developers selecionam itens e constroem o **Sprint Backlog**, seu plano para atingir esse objetivo. Durante a Sprint, eles podem adaptá-lo conforme aprendem; esclarecimentos e renegociações de escopo ocorrem com o PO sem comprometer o Sprint Goal.

O **Daily Scrum** é um evento de no máximo 15 minutos, em cada dia útil da Sprint, para os Developers inspecionarem o progresso em direção ao Sprint Goal e adaptarem o Sprint Backlog. As conhecidas três perguntas podem ser uma técnica útil, mas não são obrigatórias. A conversa pode assumir outro formato que mantenha o foco no objetivo e no plano de trabalho.

Na **Sprint Review**, o Scrum Team e stakeholders inspecionam o resultado da Sprint e discutem o que mudou no contexto. O Product Backlog pode ser adaptado a partir dessa conversa; não se trata de uma etapa formal de aceite. A **Sprint Retrospective** avalia como foi a Sprint em pessoas, interações, processos, ferramentas e Definition of Done (DoD), planejando melhorias. Em geral, é realizada após a Review e antes da próxima Planning, mas não deve ser tratada como um ritual rígido de encerramento: o importante é reservar tempo para melhoria dentro da Sprint.

Os três artefatos representam trabalho ou valor e têm compromissos associados: o **Product Backlog** se relaciona ao **Product Goal**; o **Sprint Backlog**, ao **Sprint Goal**; e o **Incremento**, à **Definition of Done**. A DoD é o entendimento compartilhado de qualidade que informa quando o Incremento é utilizável. Um pipeline verde pode compor essa definição, mas não a substitui.

△ Importante: Evite reduzir Scrum a reuniões

Eventos frequentes, ferramentas e nomes em inglês não demonstram, por si só, empirismo. Uma equipe pode usar contratos de escopo fixo ou quadros visuais por razões legítimas de contexto; o ponto é tornar restrições explícitas, inspecionar evidências e negociar mudanças de modo responsável. Evite atribuir a organizações ou casos históricos uma causa única sem fonte verificável.

Versão compilada em 20/9/2026 – Capítulo isolado do livro.

Acesse a versão completa em: https://alessiojr.com/disciplines/gps/conteudo/00-livro/

▶ Prática: GitLab como apoio ao trabalho

O GitLab pode apoiar acordos da equipe, sem "implementar Scrum" automaticamente:

1. **Product Backlog:** Issues podem registrar e ordenar necessidades, critérios e hipóteses.
2. **Cadência da disciplina:** Milestones podem representar as Sprints acadêmicas, com datas e evidências definidas pelo professor.
3. **Sprint Backlog:** Issues selecionadas e tarefas técnicas podem tornar o plano dos Developers visível; a configuração precisa permitir sua adaptação durante a Sprint.
4. **Quadro visual:** Issue Boards ou um quadro Kanban podem expor fluxo e bloqueios, mas não são artefatos nem requisitos do Scrum.

3 Perguntas reflexivas

1. "Responder a mudanças mais que seguir um plano" elimina a necessidade de planejamento e documentação? Justifique usando os valores do Manifesto.
2. Em que condições uma estratégia de capacidade e prazo estáveis, com escopo adaptável, pode ajudar? Por que ela não é uma regra universal do Scrum?
3. Se a equipe aprende no meio da Sprint que parte do plano não atende mais ao Sprint Goal, como Developers e PO podem tratar o escopo?

✓ Síntese

- O Ágil valoriza pessoas, software em funcionamento, colaboração e resposta a mudanças, sem invalidar os itens à direita.
- Scrum usa transparência, inspeção e adaptação para lidar com problemas complexos. Seus eventos são oportunidades de aprendizado, não cerimônias de prestação de contas.
- A Sprint dura no máximo um mês. O Sprint Backlog é um plano dos Developers, adaptável enquanto preserva o Sprint Goal.
- Product Owner e Developers colaboram para esclarecer e negociar escopo à luz de novas evidências; o Scrum Master apoia a efetividade desse trabalho.

Versão compilada em 20/9/2026 – Capítulo isolado do livro.

★ Checkpoint do Projeto: Projeto integrador

As escolhas a seguir são regras acadêmicas ou acordos do grupo, não requisitos adicionais do Scrum:

- Registrem o Product Backlog no GitLab e identifiquem PO, Scrum Master e Developers do grupo.
- Adotem a cadência de Sprint definida pela disciplina e explicitem o Sprint Goal e as evidências de qualidade esperadas.
- Escolham, se útil, um quadro visual e uma forma de acompanhar bloqueios. Revisem o acordo na Retrospectiva.

No próximo capítulo, aplicaremos esses princípios à escrita e à priorização de Histórias de Usuário e Épicos.

Acesse a versão completa em: https://alessiojr.com/disciplines/gps/conteudo/00-livro/

Versão compilada em 20/9/2026 – Capítulo isolado do livro.