

GPS - Gestão de Projetos de Software

Aula 12: Gestão Híbrida e Escala

Prof. Aléssio Miranda Júnior
alessio@cefetmg.br

CEFET-MG - Campus Timóteo
Dep. Engenharia de Computação

Novembro de 2026

- 1 Introdução
- 2 Teatro Ágil e o Choque de Realidade
- 3 Water-Scrum-Fall
- 4 TI Bimodal (Gartner)
- 5 Ágil em Escala (Scaled Agile)
- 6 O Tático: ScrumBan
- 7 Quando Usar Qual Metodologia?
- 8 A Prática no Projeto Integrador

- Compreender o choque entre a Agilidade e a Burocracia Financeira.
- Identificar e combater a doença do **Teatro Ágil**.
- Discutir a **TI Bimodal** (Gartner) e os sistemas Legados.
- Mapear o **Water-Scrum-Fall** e seus gargalos.
- Explorar as metodologias de Escala: **SAFe, LeSS, Nexus**.
- Reforçar a eficiência tática do **ScrumBan**.
- Entender como o GitLab atende o contexto Híbrido.

Fim do Sonho Mágico

"Se você procurar uma megacorporação rodando Scrum 'de forma pura e impecável', vai acabar morrendo de fome. A teoria colide com a gravidade de burocracias, orçamentos milionários e conselhos medrosos."

Neste décimo segundo capítulo, vamos parar de sonhar com ecossistemas perfeitos. Aceite o caos tático e aprenda a operar na **Gestão Híbrida**, soldando o controle Tradicional no motor do Ágil.

O Contexto da Gestão Híbrida

A realidade pragmática de bancos e estatais é governada pelo *Compliance* e pelo setor Financeiro.

O Manifesto Ágil pede "Respostas a mudanças". Mas o Diretor Financeiro e o TCU respondem: *"Se você não me der um escopo fixo e um orçamento fechado, não recebe um centavo"*.

Da colisão entre ****Controle**** e ****Adaptação****, nasce o modelo Híbrido.



O fenômeno mais perigoso da adoção corporativa superficial é o **Agile Theater** (Teatro Ágil).

Mudam-se os nomes, mantém-se os vícios

A diretoria renomeia o "Gerente de Projetos" para "Scrum Master", chama a "Reunião de Status" de "Daily", mas:

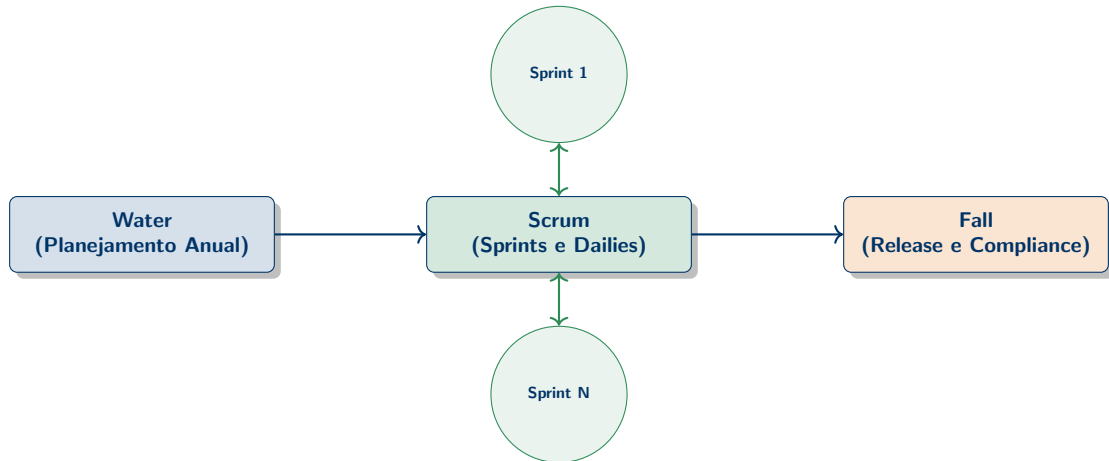
- O escopo é 100% fixado em contrato por decreto.
- A equipe não tem autonomia para rejeitar demandas absurdas.
- A Sprint é apenas um "fatiamento" artificial de um cronograma Cascata inegociável.

Como o desenvolvedor percebe que está no Teatro Ágil?

- O PO (Product Owner) nunca fala com o cliente real, é apenas um tirador de pedidos da diretoria.
- Falhar em entregar a Sprint resulta em punição verbal e e-mails agressivos, não em adaptação e retrospectiva.
- As estimativas (Story Points) são convertidas em Horas pelos gerentes e cobradas como metas absolutas.

O Frankenstein Corporativo: Water-Scrum-Fall

Muitas vezes criticado pelos puristas, mas amplamente executado. Ele isola a agilidade no "miolo" da engenharia, mantendo as extremidades ancoradas na burocracia do Cascata.



1. **WATER (Planejamento e Orçamento):** Durante meses, a diretoria define orçamento e arquitetura de longo prazo. O escopo é congelado no jurídico.
2. **SCRUM (Desenvolvimento):** A equipe técnica atua de forma isolada e ágil, dividindo as fatias em Sprints e validando o código em CI/CD. A engenharia respira!
3. **FALL (Governança e Entrega):** O software bate de frente no comitê de segurança (CAB), auditoria manual e regras de mudança. Demora meses para entrar no ar.

O Paradoxo do Gargalo (Fall)

O problema estrutural do Water-Scrum-Fall reside na sua ponta final.

A Agilidade Morre na Praia

Mesmo que a equipe de Devs rode ciclos maravilhosos de 2 semanas (Integração Contínua), se a auditoria e aprovação jurídica levarem 3 meses, o cliente final só recebe valor a cada trimestre. **A entrega contínua (CD) foi estrangulada.**

O Gartner criou esse termo para explicar o ecossistema nas grandes empresas, que dividem a T.I. em dois motores:

- **Modo 1 (Tradicional):** Lento, confiável, focado em alta segurança para "sistemas de registro" e legado. (Ex: O banco de dados central que paga sua bolsa ou processa notas há 25 anos).
- **Modo 2 (Ágil):** Rápido, inovador, focado em experimentação iterativa (Startups Internas). (Ex: Um novo App em React Native ou um Agente LLM).

A habilidade do Gestor moderno é **orquestrar os dois modos em harmonia**.

Tutor IA conectado ao SUAP

Sua equipe cria a interface do robô em ritmo frenético toda semana (Modo 2 - Ágil). Mas para saber se o aluno está reprovado, o robô consome uma API do SUAP (Modo 1 - Legado e Lento). Não tem milagre: você vai ter que gerenciar esse atraso e a burocracia governamental.

O *Scrum Guide* foi pensado para **uma única equipe de até 9 pessoas**.

O que acontece com a Nubank, o Mercado Livre ou o Itaú, que possuem 500, 1.000 ou 5.000 engenheiros programando simultaneamente o mesmo produto?

A resposta: **Ágil em Escala**. Surgem *Frameworks* gigantes para forçar a sincronia das equipes.

O **SAFe (Scaled Agile Framework)** é o rei do mercado corporativo (mais de 70% de adoção). Ele divide a empresa em camadas:

- **Nível de Equipe:** Devs rodando Scrum e Kanban comum.
- **Nível de Programa (ART - Agile Release Train):** O "Trem Ágil" junta 5 a 12 equipes. Elas sincronizam a cada 10 semanas no mega evento **PI Planning**.
- **Nível de Portfólio:** Onde os diretores enviam Épicos Multimilionários.

O **PI Planning** paralisa 150 engenheiros em uma sala (ou Zoom) por 2 dias seguidos. Eles conectam dependências em quadros visuais.

SAFe: Cascata Turbinada?

Muitos arquitetos odeiam o SAFe. Eles acusam o framework de ser extremamente prescritivo, focar demais na gerência, cobrar certificações caras e matar a agilidade orgânica. Porém, a defesa é: governar 500 devs sem uma burocracia padronizada geraria um caos muito pior.

Como sincronizar 15 equipes diferentes que trabalham no mesmo software?

- **O Program Board:** Durante o evento, cria-se um mural gigante.
- Se a Equipe A (App Mobile) precisa que a Equipe B (Banco de Dados) crie uma API antes, os desenvolvedores conectam os cartões das duas equipes com um fio de lã física (ou seta virtual).
- Se a Equipe B diz "Não dá tempo", a diretoria está lá na sala para decidir quem ganha a prioridade em tempo real.

Se o SAFe é um canhão pesado, temos alternativas.

O **LeSS (Large-Scale Scrum)** adota a filosofia "Mais Scrum, Menos Framework".

- **A Regra de Ouro:** *Um único* Product Backlog para até 8 times diferentes trabalhando no mesmo produto.
- Força a diretoria e o Product Owner a fazerem uma priorização radical em escala, em vez de isolar cada time no seu próprio "mundinho".

Criado por Ken Schwaber (o pai do Scrum Original).

- Foca puramente na **Integração de Código**.
- Adiciona apenas um novo artefato (o *Nexus Sprint Backlog*).
- Cria a figura da **Equipe de Integração** (Nexus Integration Team), cujo único objetivo não é codificar novas **features**, mas sim garantir que o código das 9 equipes rode junto sem travar o servidor.

No dia a dia operacional, equipes de Sustentação sofrem no Scrum puro. Se o sistema de banco de dados cair às 3h da manhã, o time de infraestrutura vai esperar a "Sprint Planning" da semana que vem para consertar?

A solução é o ScrumBan:

- **Fluxo Contínuo:** Não existe Sprints fechadas. A prioridade muda no dia.
- **Rituais Scrum mantidos:** Daily e Retrospectivas quinzenais continuam, garantindo que o time converse e não vire um grupo isolado de tarefeiros.

- O **PO** (Gestor) pode injetar um incidente *Pri-1* a qualquer minuto do dia no topo do Backlog (diferente do Scrum purista).
- Mas a proteção do Kanban (**Limite de WIP**) proíbe os Devs de começarem 10 bugs de uma vez só.
- O foco vira totalmente **Vazão** (*Throughput*) e **Tempo de Resolução** (*Lead Time*).

Princípio da Mínima Cerimônia

Nunca seja dogmático. Usar SAFe numa startup de 4 amigos é assassinato de produtividade. Escolha o chassi adequado:

Cenário	Modelo e Justificativa
Startup, MVP (5–9 Devs)	Scrum / XP Ciclos curtos, alto feedback e escopo livre.
Infraestrutura e Bugs (Ops)	ScrumBan Fluxo contínuo com WIP; emergências a qualquer hora.
Corporação Leve (3–9 Times)	Nexus / LeSS Sincronia limpa sem papéis gerenciais extras.
Bancão, +50 devs, TCU	SAFe ou Water-Scrum-Fall Necessidade violenta de <i>compliance</i> e orçamentos jurídicos.

Plataformas maduras de engenharia (*GitLab*, *Azure DevOps*) suportam qualquer escolha metodológica da sua equipe:

- Quer rodar Scrum corporativo? Crie *Milestones* para as Sprints e *Releases* fechadas.
- Quer rodar Kanban/ScrumBan? Ative o **Issue Board**, desconsidere o Milestone, arraste as colunas e obedeça ao Limite Virtual.
- Quer escalar? Use o sistema de *Épicos* e Sub-Issues.

Na reta final do projeto, vocês usarão essa flexibilidade a favor do grupo:

Se no Check 04 a equipe estiver desesperada com bugs ou com problemas críticos de servidor na semana de entrega, **cortem a cerimônia**. Mudem taticamente para ScrumBan, parem de adicionar novas features (limitem o WIP drasticamente) e façam o **Swarming** para entregar o que já está na esteira.

1. Por que o setor Financeiro rejeita propostas de "Ágil Puro" e força o modelo *Water-Scrum-Fall*?
2. Como a "Fase 3" (Fall) destrói as vantagens e os lucros táticos da Integração Contínua desenvolvida na Fase 2?
3. Se a sua equipe gerencia servidores e atende 20 tickets de queda por semana, por que a adoção de Sprints fechadas do Scrum é fatal?

Adaptação é Sobrevivência

Email: alessio@cefetmg.br

Horário de aula: Segunda, 7h às 8:40h

Horário de atendimento: Quarta 16h e whatsapp

Preparem o GitLab!

Próxima aula: Mergulharemos nas engrenagens das Ferramentas de Gestão Modernas (GitLab x Jira).