

BIUD Talks



Mindset ágil

Santiago

Cronograma do Programa



22/04

Wanderson

Inteligência Artificial para Dev



29/04

Praxedes

Testes Unitários



06/05

Luk

Arquitetura de Software



13/05

Regis

Integração de Sistemas



20/05

Rogério

Root Cause Analysis (RCA)



27/05

Gustavo

Metodologia Ágil



03/06

Luk

CR – Code Review

Cronograma do Programa

10/06	Thiago	IA na prática: Claude Code
17/06	Guilherme	Segurança de dados no ambiente corporativo e pessoal
24/06	Jean	Machine Learning
01/07	Santiago	CNV - Comunicação não Violenta no ambiente de trabalho
08/07	Rogério	Saúde Mental
15/07	Guilherme	DevOps e Infraestrutura
22/07	Santiago	Mindset ágil

Cronograma do Programa

29/07	Gustavo	Ambidestria Organizacional
12/08	Wanderson	Como uma demanda nasce, atravessa as áreas e vira valor para o cliente
19/08	Anderson	Busca Inteligente de Dados na Jornada ESG: encontrando empresas, evidências e oportunidades por similaridade
26/08	Luk	Banco Vetorial
02/09	Alex	Teste unitário

Por que isso importa?



A maioria das startups não morre por falta de código. Morre por falta de foco no que gera valor.

42%

criaram algo que o mercado
não queria

29%

ficaram sem caixa
antes de validar o valor

23%

tinham o time errado
para aprender rápido

Fonte: CB Insights, "The Top 12 Reasons Startups Fail", 2021

Não é um framework



Frameworks e rituais

- Scrum, Kanban, XP
- Sprint, daily, board
- Story points, cerimônias
- São o "como", ferramentas do dia a dia



Mindset ágil

- Valorizar pessoas e colaboração
- Entregar valor de forma contínua
- Aprender rápido com o real
- É o "porquê", a atitude por trás de tudo

O framework sem o mindset é só teatro.

As origens: o Manifesto Ágil (2001)



17 desenvolvedores se reuniram em Utah e resumiram décadas de aprendizado em 4 valores.

1

Indivíduos e interações

mais que processos e ferramentas

2

Software funcionando

mais que documentação abrangente

3

Colaboração com o cliente

mais que negociação de contratos

4

Responder a mudanças

mais que seguir um plano

Princípios mais valiosos



Dos 12 princípios do Manifesto, estes são os que mais mudam o dia a dia de um time enxuto:

1

Entrega contínua e frequente

Nossa prioridade é entregar valor de forma cedo e constante, não em um grande lançamento só.

2

Simplicidade

A arte de maximizar o trabalho não feito é essencial, fazer menos, e melhor.

3

Aceitar mudanças

Mudar requisitos, mesmo tarde, deve virar vantagem competitiva, não um problema.

4

Times auto-organizados

As melhores arquiteturas e decisões emergem de times com autonomia real.

Cada feature é uma hipótese, não uma certeza



O ciclo "Construir → Medir → Aprender" é o motor do Lean Startup.



Na prática: antes de construir a feature inteira, perguntamos "qual é a menor coisa que testa essa hipótese de valor?"

MVP: o menor experimento que ensina



O que NÃO é

- Uma versão "capenga" e incompleta da ideia final
- Um produto sem qualidade só pra sair rápido
- Desculpa pra pular teste e planejamento



O que É

- O menor conjunto de funcionalidades que gera aprendizado real
- Um experimento com uma hipótese clara de valor
- O ponto de partida de um ciclo de aprendizado validado

Resolva a dor primeiro, sofisticação depois



Sem resolver o básico, o encantador não importa pra ninguém.

Básico (must-be)

Se falhar aqui, o cliente abandona. Não importa o resto

Desempenho

Quanto melhor, mais satisfação. Mas só depois do básico resolvido

Encantador (delighter)

O "uau" que só faz sentido em cima de uma base sólida

Valor de verdade: resultado, não volume



A "armadilha da construção" é medir sucesso pelo que entregamos, não pelo que mudou pro cliente.

Output (o que fizemos, saídas)	Outcome (o que mudou, resultados)
"Lançamos 10 features este trimestre"	"Reduzimos o cancelamento em 15%"
"Fechamos todos os cards do sprint"	"Aumentamos ativação de novos usuários"
"Entregamos no prazo"	"O cliente resolveu o problema dele"

Simples não é sinônimo de mal feito



Pode simplificar	Nunca é negociável
Número de telas e passos do fluxo	Bugs conhecidos em produção
Quantidade de opções e configurações	Falhas de segurança
Casos de uso raros e hipotéticos	Validação de dados do usuário
Nível de polimento visual do MVP	Cobertura mínima de testes combinada

Fontes: Schwaber, K.; Sutherland, J. *The Scrum Guide* — conceito de *Definition of Done*, [scrumguides.org](https://www.scrumguides.org), 2020 · Fowler, M. "*TechnicalDebtQuadrant*", martinfowler.com, 2009

Overengineering



A gente entrega em ciclos contínuos. Cada camada extra que ninguém pediu é tempo a menos pra próxima entrega de valor.

?

O instinto do dev

- Adicionar camadas "pra garantir flexibilidade futura"
- Tratar casos extremos hipotéticos que talvez nunca aconteçam
- Construir algo genérico e configurável quando bastava algo direto
- "Já que estou aqui, deixo preparado pra X também"

Y

YAGNI - You Aren't Gonna Need It

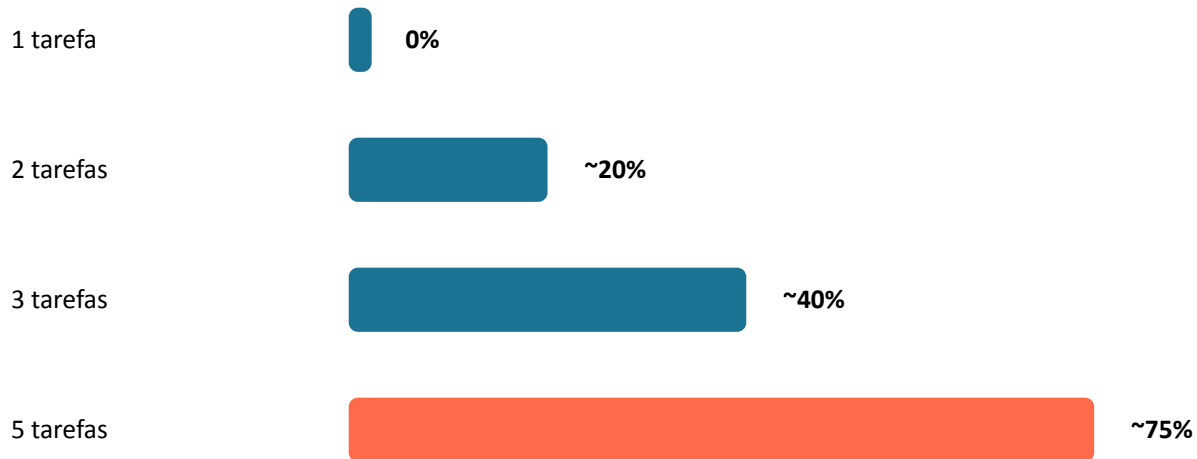
- Construa só o que resolve o pedido de agora
- Simplicidade não é preguiça, é disciplina
- Complexidade especulativa custa tempo sem valor comprovado
- "Gold plating": extra não pedido = custo sem valor percebido



Multitarefa nem sempre é produtividade

Tarefas em paralelo

Tempo perdido em troca de contexto



Práticas para refletirmos



1

Resolva a dor antes de sofisticar

Pergunte "isso já tira a dor do cliente?" antes de "isso está completo?".

2

Aplique o teste do YAGNI ("You Aren't Gonna Need It")

Antes de generalizar, pergunte: "alguém pediu isso, ou é um 'e se' meu?"

3

Feito > perfeito, com Definition of Done

Simplificar escopo, sim. Pular teste, revisão de segurança ou erro, não.

4

Rode o checklist de qualidade antes do "Pronto"

Zero bugs conhecidos e zero falhas de segurança, mesmo na versão mais simples.

5

Comemore simplicidade tanto quanto sofisticação

Reconheça quando alguém resolve com menos, não só quando resolve com mais.

Abrindo o Debate



Questões para reflexão:

1. Qual foi o principal aprendizado para você?
2. Como você pode aplicar isso em seu dia a dia?
3. Quais foram as dúvidas geradas?
4. Que conexões você vê com outros tópicos?

Obrigado!

Até o próximo episódio da BIUD Talks



Inovando juntos |



Crescendo juntos