

ENGENHARIA DE SOFTWARE

Métodos de desenvolvimento ágeis

HISTÓRICO / MOTIVAÇÃO

➤ 2000s: engenharia de software ágil:

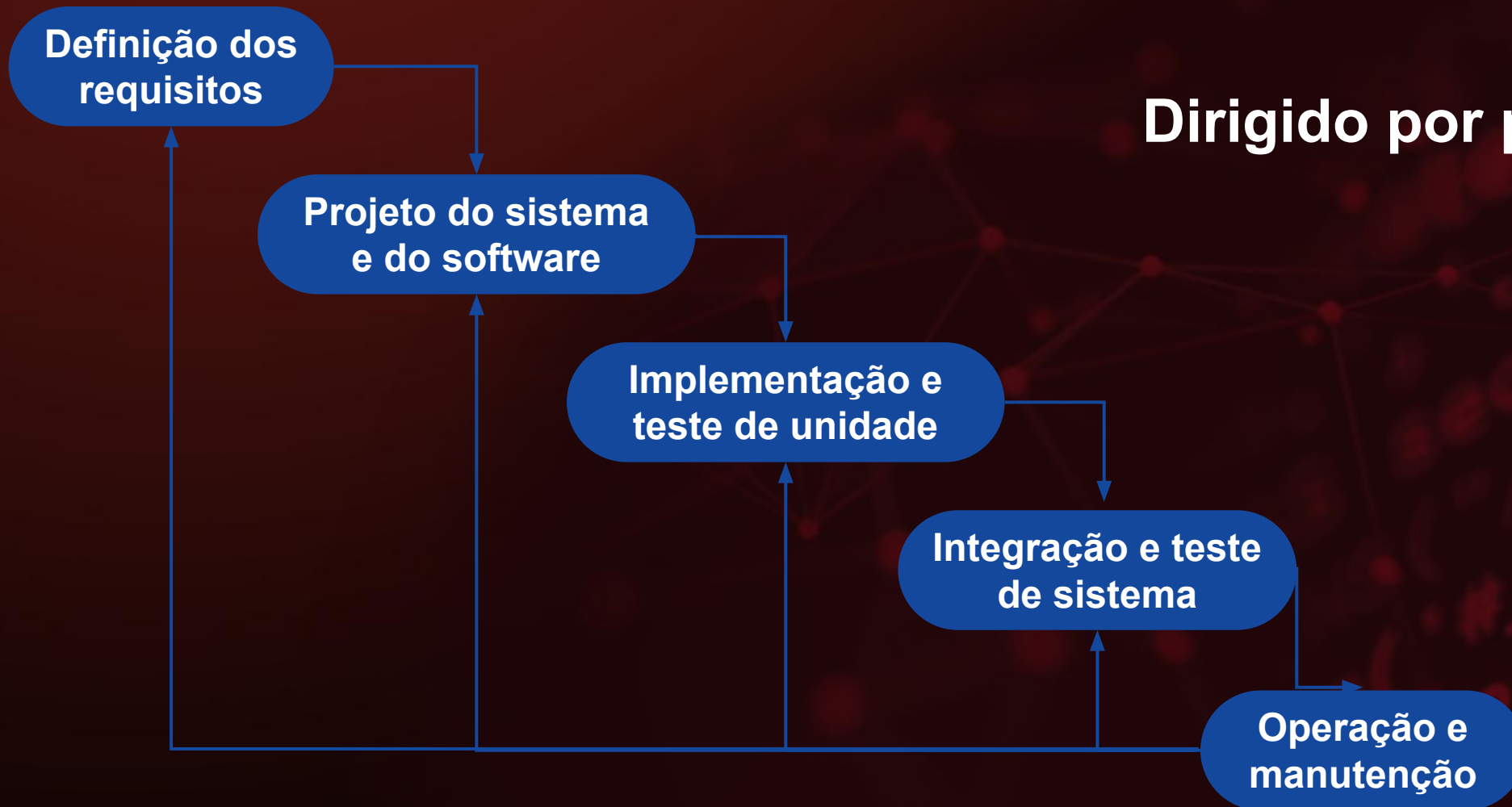
- ✓ Ambientes corporativos menores (médio e pequeno porte)
- ✓ Processos baseados em planos causam muito *overhead*.
- ✓ Gasta-se muito com planejamento/análise comparando com implementação/teste.
- ✓ No final da década de 1990, desenvolvedores propuseram processos mais “leves”: os “métodos ágeis”.
- ✓ Objetivo: produzir rapidamente softwares úteis, por meio de incrementos, com cada incremento incluindo novas funcionalidades.

DIRIGIDO POR PLANOS X ÁGEIS



DIRIGIDO POR PLANOS X ÁGEIS

Dirigido por planos



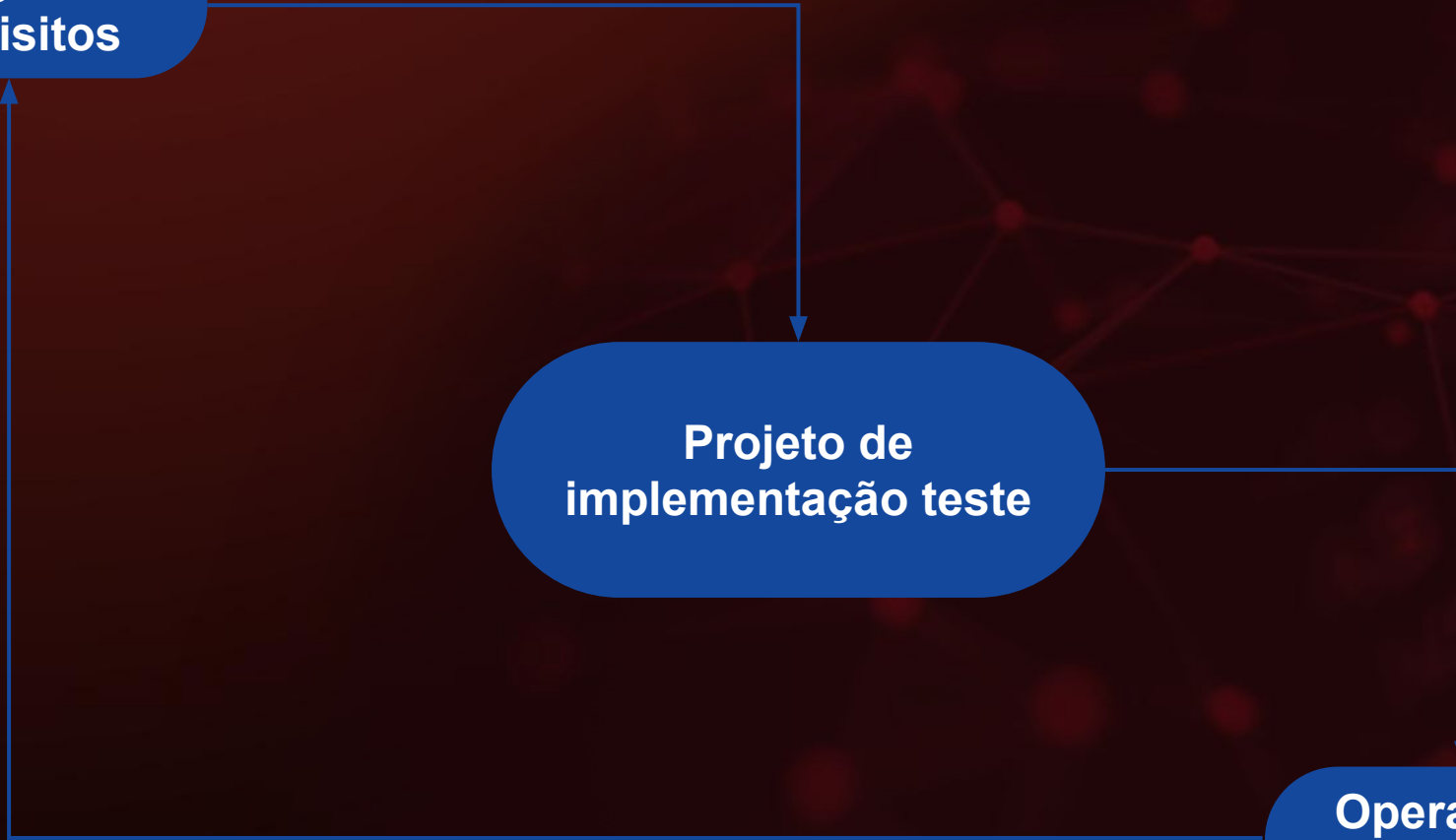
DIRIGIDO POR PLANOS X ÁGEIS

Definição dos
requisitos

Projeto de
implementação teste

Operação e
manutenção

Ágil



MANIFESTO ÁGIL, 2001

“Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo. Através deste trabalho, passamos a valorizar:

- ✓ **Indivíduos e interações** mais que processos e ferramentas
- ✓ **Software em funcionamento** mais que documentação abrangente
- ✓ **Colaboração com cliente** mais que negociação de contratos
- ✓ **Responder às mudanças** mais que seguir um plano

Ou seja, mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.”

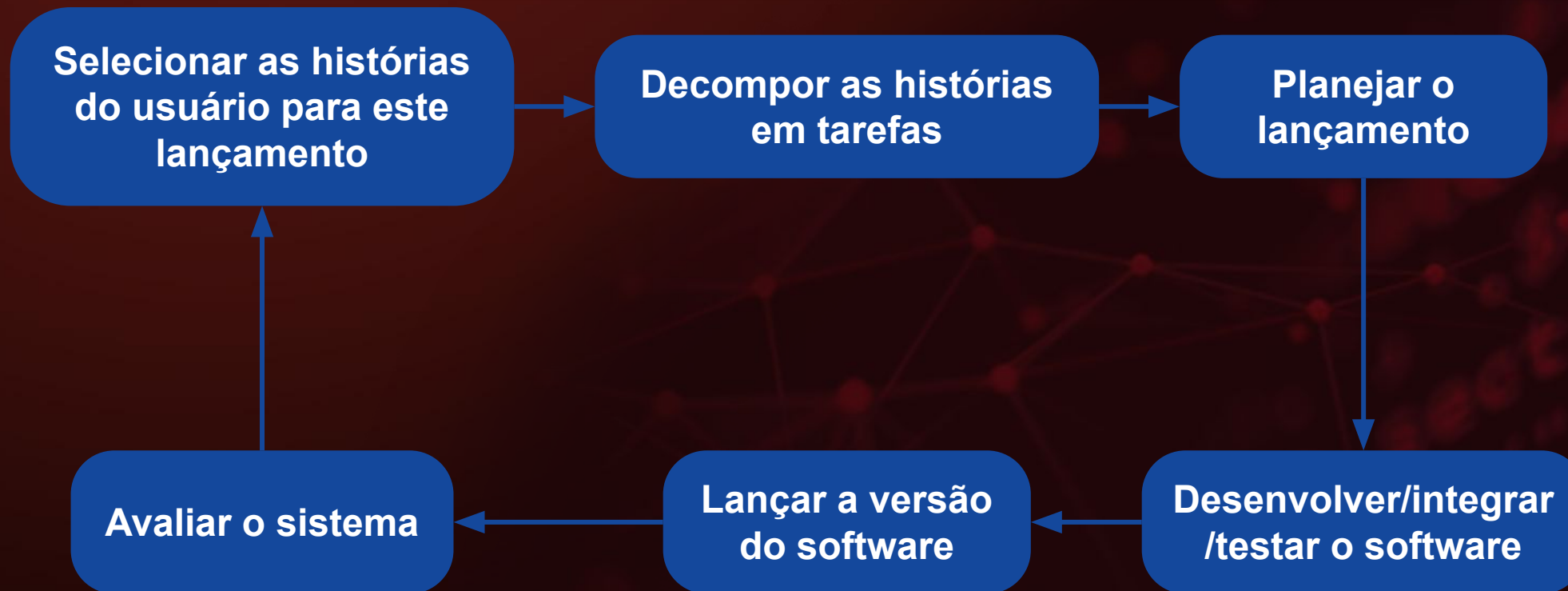
PRINCÍPIOS DOS MÉTODOS ÁGEIS

Princípio	Descrição
Envolvimento do Cliente	Os clientes devem ser envolvidos em todo o processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos de sistema e avaliar as iterações do sistema.
Acolher as mudanças	Tenha em mente que os requisitos do sistema mudam e, portanto, deve-se projetar o sistema para acomodar essas mudanças.
Entrega incremental	O software é desenvolvido em incrementos e o cliente especifica os requisitos incluídos em cada um deles.
Manter a simplicidade	Deve-se ter como foco a simplicidade, tanto do software que está sendo desenvolvido quanto do processo de desenvolvimento. Sempre que possível, trabalhe ativamente para eliminar a complexidade do sistema.
Pessoas, não processos	As habilidades do time de desenvolvimento devem ser reconhecidas e aproveitadas da melhor maneira possível. Seus membros devem ter liberdade para desenvolver seu modo próprio de trabalhar sem se prender a processos determinados.

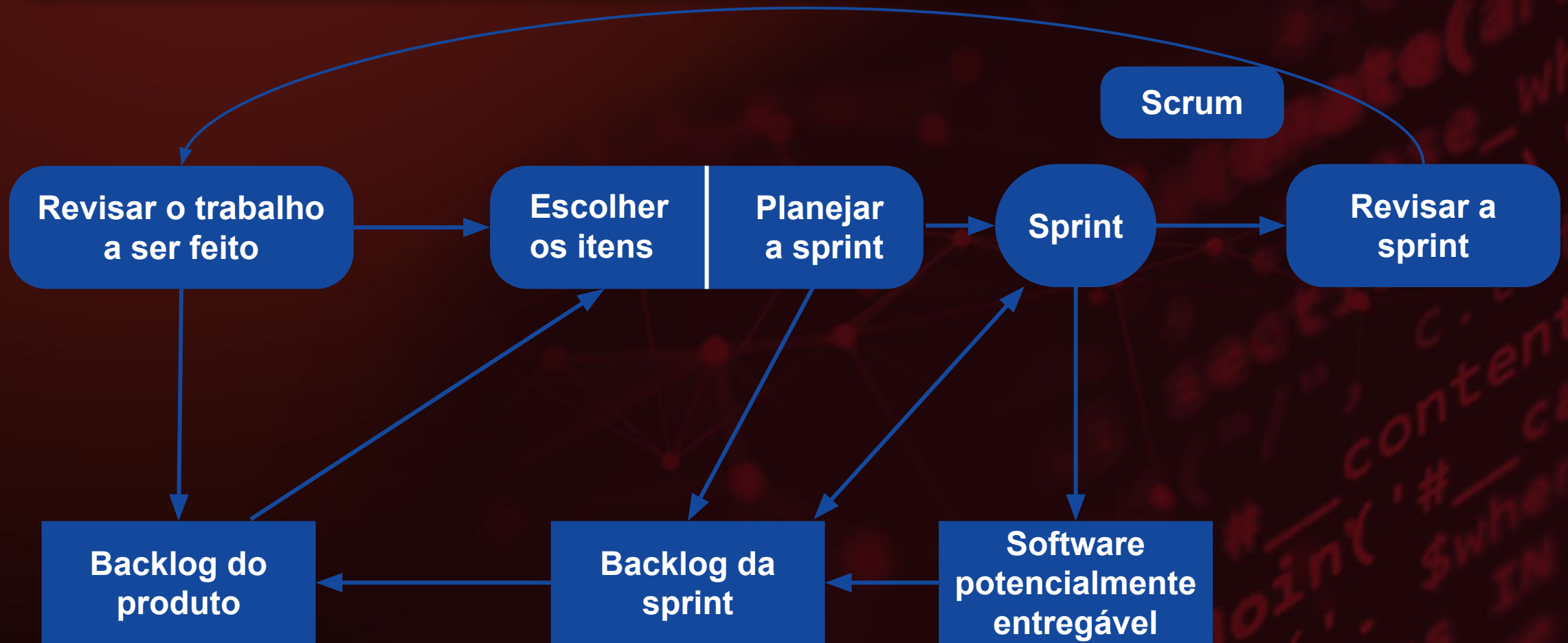
EXEMPLOS DE MÉTODOS ÁGEIS

- ✓ Extreme Programming (XP)
- ✓ Scrum
- ✓ Test Driven Development (TDD)
- ✓ Crystal
- ✓ Adaptative Software Development (ASD)
- ✓ Feature Driven Development (FDD)
- ✓ Dynamic Systems Development Method (DSDM)
- ✓ Agile Unified Process

MODELO DE CICLO DE LANÇAMENTO DO XP



MODELO DE SPRINT DO SCRUM



DIFICULDADES / LIMITAÇÕES

Cliente deve estar disposto e capaz de passar o tempo com a equipe de desenvolvimento.

Cliente deve ser capaz de representar todas as partes interessadas.

Membros da equipe podem não ter a personalidade adequada.

**A organização pode não ter a cultura adequada.
Organizações investiram muito em definição e organização de processos.**

DIFICULDADES / LIMITAÇÕES

Priorizar mudanças pode ser extremamente difícil, principalmente se há muitas partes envolvidas.

Manter simplicidade pode ser complicado. Pode-se levar tempo para se fazer as simplificações desejáveis.

Pode dificultar negociações contratuais (incluindo terceirizações).

Depende de maturidade de desenvolvedores.

REFERÊNCIAS

- **SOMMERVILLE, I. Engenharia de Software. 10. ed. São Paulo: Pearson Education do Brasil, 2018.**