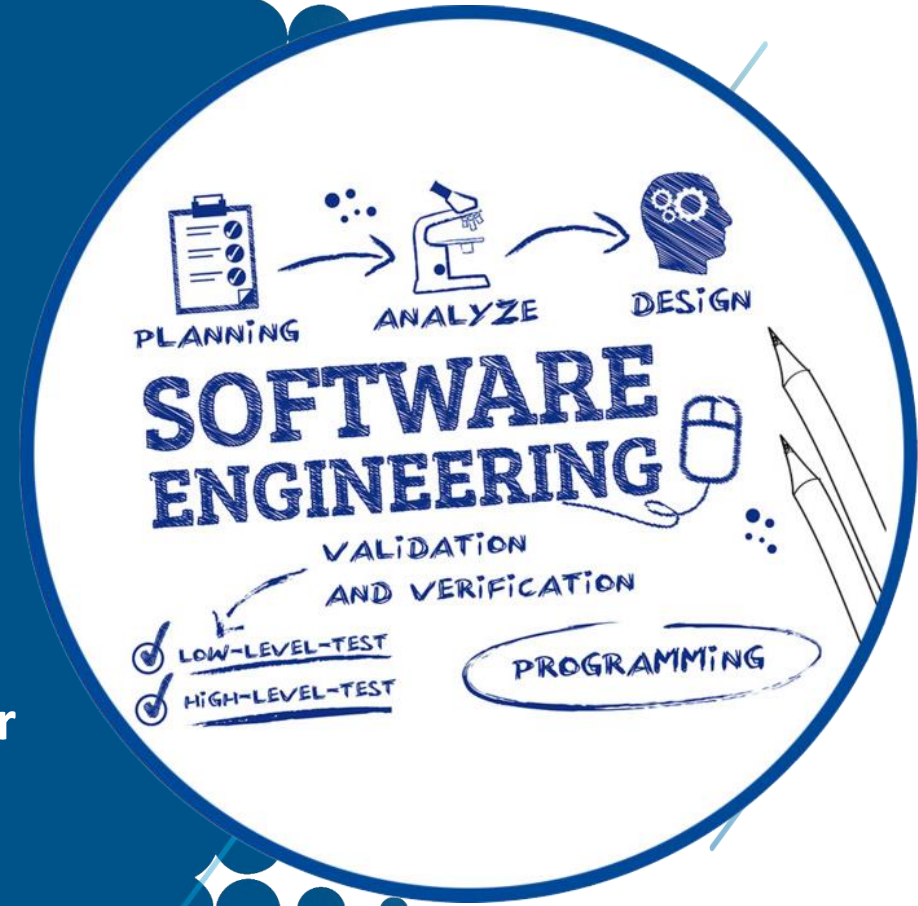


Engenharia de Software I

Prof^a. Me. Cynara Leão Garcia
cynara.garcia@unicesumar.edu.br



Programação Extrema (XP)



Programação Extrema (XP)

**Metodologia de desenvolvimento de Software
aperfeiçoada nos últimos 10 anos.**



Programação Extrema (XP)

- É uma metodologia ágil para equipes pequenas e médias e que irão desenvolver software com requisitos vagos e em constante mudança.
- A codificação é a atividade principal.
- Ênfase
 - Menor em processos formais de desenvolvimento.
 - Maior na disciplina rigorosa.

Programação Extrema (XP)

- Ideia

- Baseia-se na revisão permanente do código, testes freqüentes, participação do usuário final, refatoramento contínuo, integração contínua, planejamento, design e redesign a qualquer hora.

- Usuários

- Ford.
 - BMW.
 - Symantec.

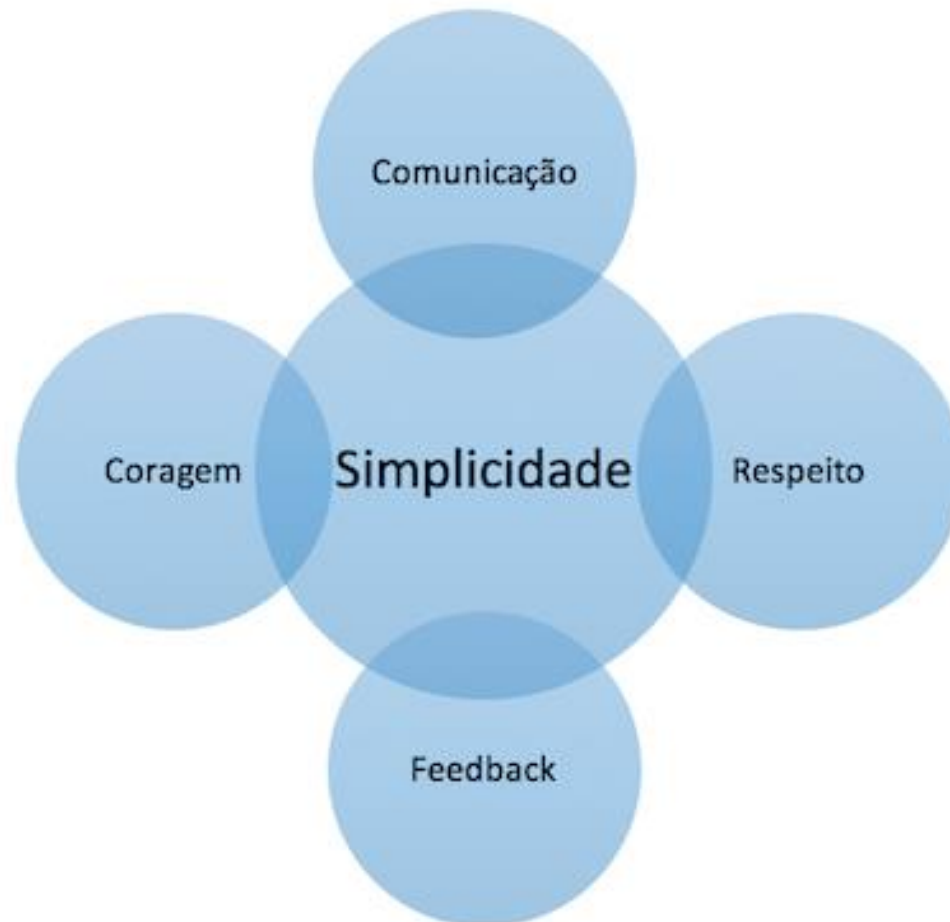
Princípios Básicos (XP)

- Feedback rápido
- Simplicidade é o melhor negócio
- Mudanças incrementais
- Carregue a bandeira das mudanças / não valorize o medo
- Alta qualidade do código

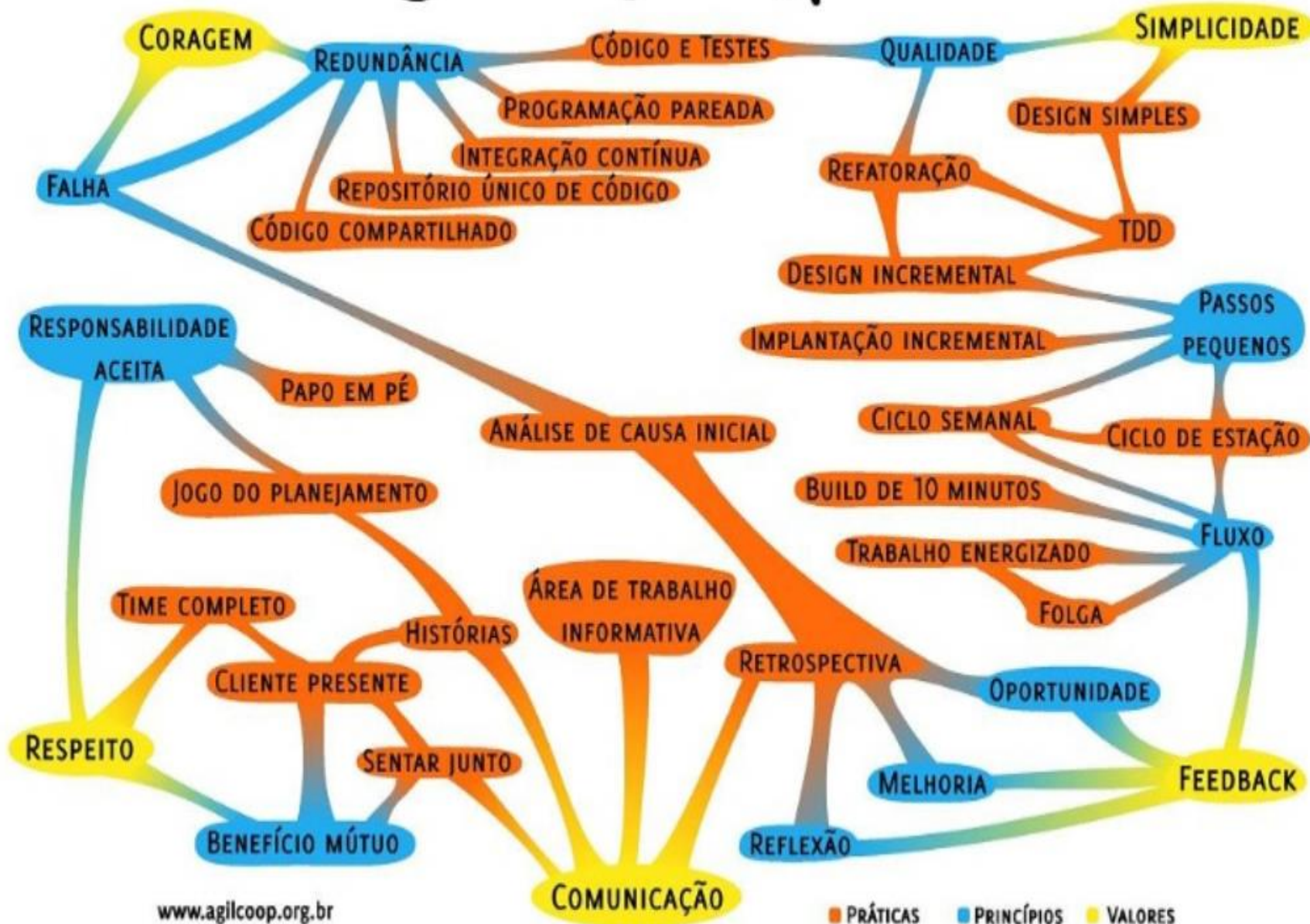


Programação Extrema (XP)

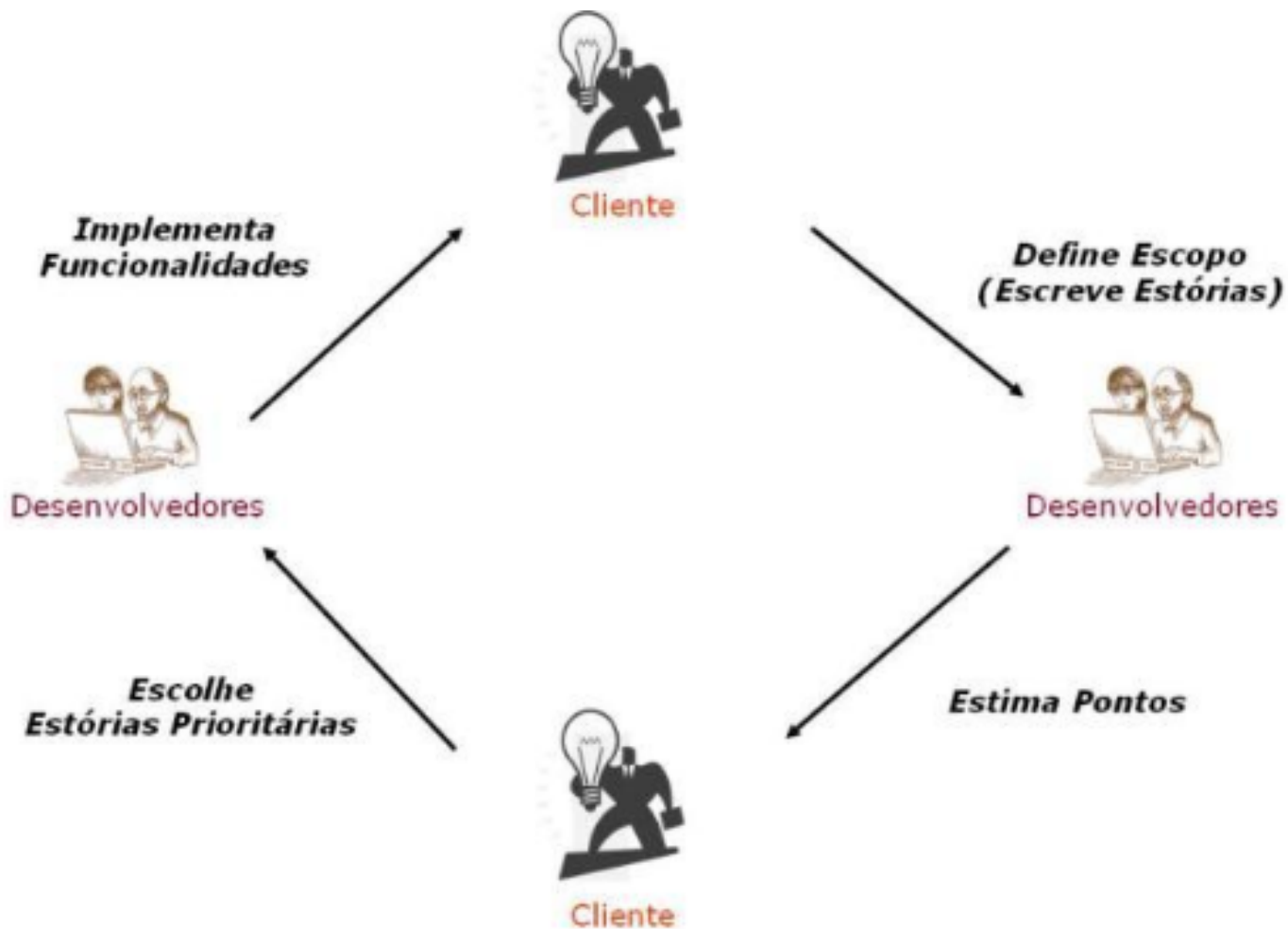
- Os valores da XP



Programação extrema



XP – Ciclo de Vida



Programação Extrema (XP) - Práticas

As 12 práticas de XP

- | | |
|-------------------|-------------------------------|
| 1. Planejamento | 7. Programação em Pares |
| 2. Fases Pequenas | 8. Propriedade Coletiva |
| 3. Metáfora | 9. Integração Contínua |
| 4. Design Simples | 10. Passo Sustentável |
| 5. Testes | 11. Cliente + Desenvolvedores |
| 6. Refatoramento | 12. Padronização do Código |

Práticas de XP

– Prática XP na qual se define

- estimativas de prazo para cada tarefa.
- as prioridades: quais as tarefas mais importantes.

– Dois passos chave:

• Planejamento de um release

- Cliente propõe funcionalidade desejadas (estórias).
- Programadores avaliam a dificuldade de implementá-las.

• Planejamento de uma iteração (até 3 semanas)

- Cliente define as funcionalidades prioritárias para a iteração.
- Programadores as quebram em tarefas e avaliam o seu custo.

Práticas de XP

– Prática XP na qual se define

- estimativas de prazo para cada tarefa.
- as prioridades: quais as tarefas mais importantes.

– Dois passos chave:

- **Planejamento de um release**

- Cliente propõe funcionalidade desejadas (estórias).
- Programadores avaliam a dificuldade de implementá-las.

Práticas de XP

- **Planejamento de uma iteração (até 3 semanas)**

- Cliente define as funcionalidades prioritárias para a iteração.
- Programadores as quebram em tarefas e avaliam o seu custo.
- Ótimo feedback para que o cliente possa dirigir o projeto
 - É possível ter uma ideia clara do avanço do projeto.
 - Clareza reduz riscos, aumenta a chance de sucesso.

- **Testes de aceitação**

- No jogo de planejamento, o usuário-cliente elabora histórias que descrevem cada funcionalidade desejada. Programador as implementa.
 - Cada história deve ser entendida suficientemente bem para que programadores possam estimar sua dificuldade.
 - Cada história deve ser testável.

Práticas de XP

- **Testes de aceitação são elaborados pelo cliente**

- São testes automáticos.
- Quando rodarem com sucesso, a funcionalidade foi implementada.
- Devem ser rodados novamente em cada iteração futura.
- Oferecem feedback: pode-se saber, a qualquer momento, quantos % do sistema já foi implementado e quanto falta.

Práticas de XP

- **Design simples**

- O design está presente em todas as etapas do XP
 - O projeto começa simples e se mantém simples através de testes e refinamento do design (refatoramento).
- Todos buscamos design simples e claro. Em XP, levamos isto a níveis extremos.
 - Não permitimos que se implemente nenhuma função adicional que não será usada na atual iteração.

Práticas de XP

- **Programação em pares**

- Todo o desenvolvimento em XP é feito em pares

- Um computador, um teclado e dois programadores.

- Um piloto, um co-piloto.

- Papéis são alternados frequentemente.

- Pares são trocados periodicamente.

Práticas de XP

– Benefícios da programação em pares

- Melhor qualidade do design, código e testes.
- Revisão constante do código.
- Nivelamento da equipe.
- Maior comunicação.



Práticas de XP

● Testes

- O seu desenvolvimento é guiado por testes
- Os testes puxam o desenvolvimento.
- Os programadores XP escrevem testes primeiro, escrevem código e rodam testes para validar o código escrito.
- Cada unidade de código só tem valor se seu teste funcionar 100%.
- Todos os testes são executados automaticamente, o tempo todo.
- Testes são a documentação executável do sistema.

Práticas de XP

- Testes dão maior segurança: coragem para mudar
 - Que adianta a OO isolar a interface da implementação se o programador tem medo de mudar a implementação?
 - Código testado é mais confiável.
 - Código testado pode ser alterado sem medo.



Práticas de XP

- **Refinamento do design (refatoramento)**

- Não existe uma etapa isolada de design em XP
 - O código é o design.
- O design é melhorado continuamente através de refatoramento.
- O refatoramento é um processo formal realizado através de etapas reversíveis
 - Passos de refatoramento melhoram, incrementalmente, a estrutura do código, sem alterar sua função.
 - Existência prévia de testes é essencial (elimina o medo de que o sistema irá deixar de funcionar por causa da mudança).

Práticas de XP

● Integração contínua

- Projetos XP mantêm o sistema integrado o tempo todo
 - Integração de todo o sistema pode ocorrer várias vezes ao dia.
 - Todos os testes (unidade e integração) devem ser executados.
- Integração contínua reduz o tempo passado no inferno da integração.

Práticas de XP

● Posse coletiva

- Em um projeto XP, qualquer dupla de programadores pode melhorar o sistema a qualquer momento.
- Todo o código em XP pertence a um único dono: a equipe
 - Todo o código recebe a atenção de todos os participantes resultando em maior comunicação.
 - Maior qualidade (menos duplicação, maior coesão).
 - Menos riscos e menos dependência de indivíduos.
- Todos compartilham a responsabilidade pelas alterações

Práticas de XP

● Padrões de codificação

– O código escrito em projetos XP segue um padrão de codificação, definido pela equipe

- Padrão para nomes de métodos, classes, variáveis.

- Organização do código.

– Todo o código parece que foi escrito por um único indivíduo, competente e organizado.

– Código com estrutura familiar facilita e estimula

#Posse coletiva #Comunicação mais eficiente #Simplicidade

#Programação em pares #Refinamento do design

Práticas de XP

- **Metáfora**

- Equipes XP mantêm uma visão compartilhada do funcionamento do sistema.
 - Pode ser uma analogia com algum outro sistema que facilite a comunicação entre os membros da equipe e cliente.
- Facilita a escolha dos nomes de métodos, classes, campos de dados etc.
 - Serve de base para estabelecimento de padrões de codificação.

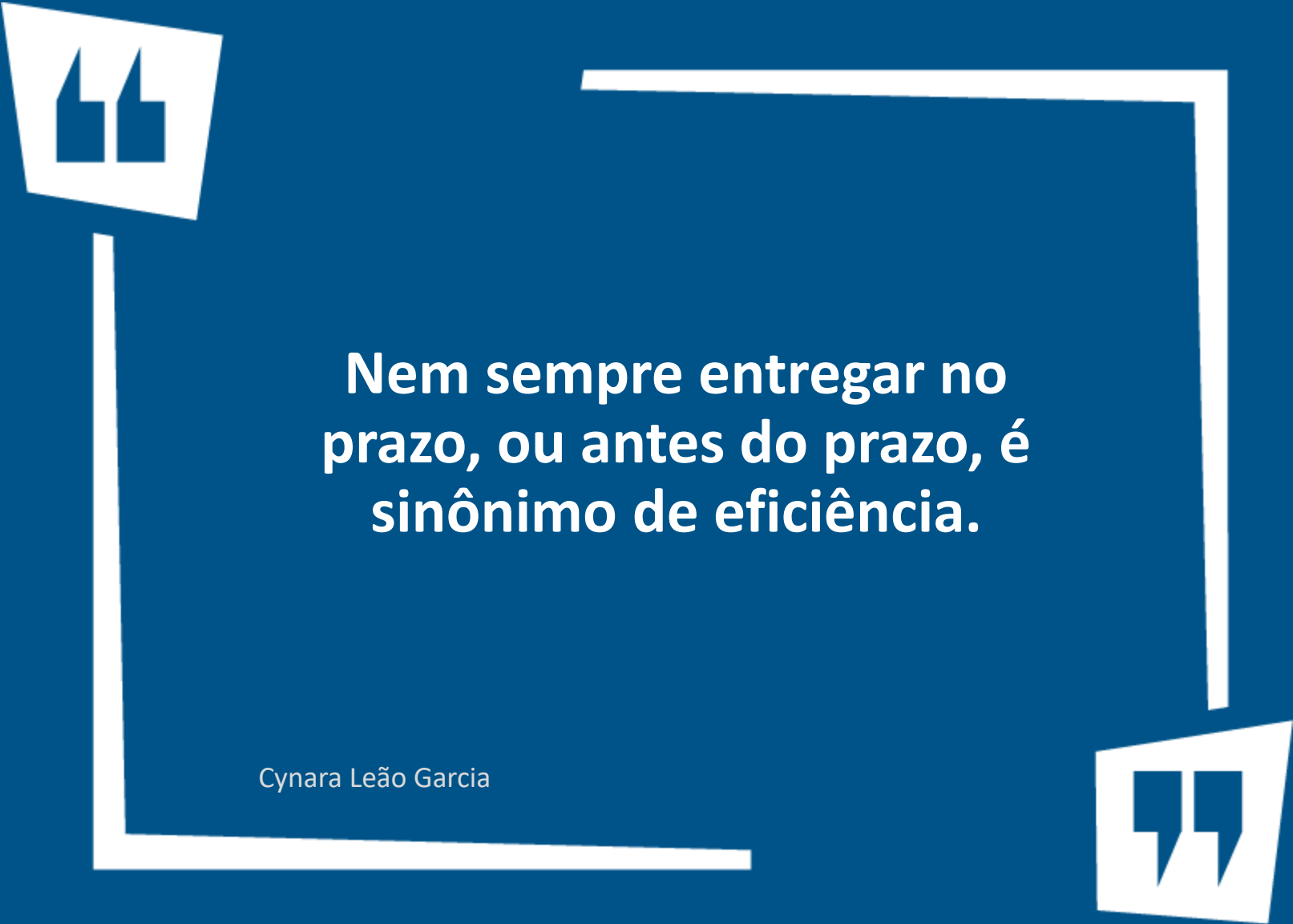
Quando não Usar XP

- Equipes grandes e espalhadas geograficamente
- Situações onde não se tem controle sobre o código
- Situações onde o feedback é demorado



Atividade





**Nem sempre entregar no
prazo, ou antes do prazo, é
sinônimo de eficiência.**

Cynara Leão Garcia

