Programação Extrema e outras práticas para elaboração de software

Hugo Corbucci
AgilCoop
DCC - IME - USP
www.agilcoop.org.br

AgilCoop

O que é Programação Extrema?

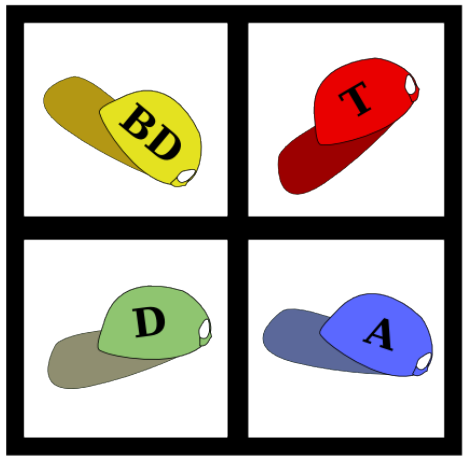
- Conjunto de práticas a serem adotadas no cotidiano da equipe.

  As práticas:
  - são adaptáveis a diferentes contextos
  - se suportam e complementam
  - permitem adoção em pequenos passos
  - apoiam os valores essenciais por trás da metodologia

Os Valores de XP

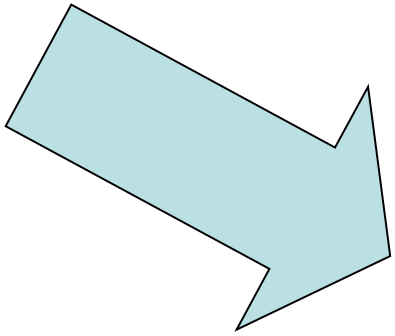
- Comunicação
- *Feedback*
- Coragem
- Simplicidade
- Respeito

A equipe e seu ambiente

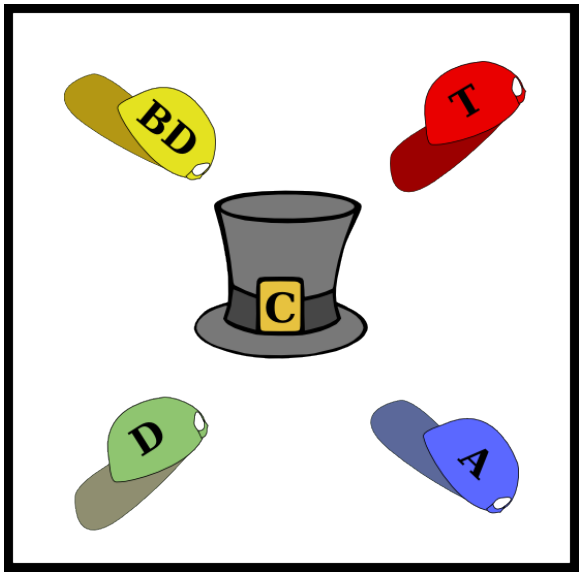


Time completo

Cliente presente



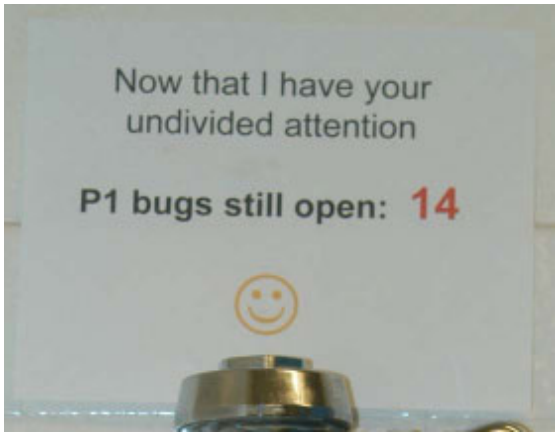
Sentar juntos



A equipe e seu ambiente

- Papéis
  - *Coach*: Lembra a todos as práticas e ajuda com dificuldades na equipe
  - *Tracker*: Mantem informações sobre o projeto e elabora gráficos que mostrem as mais importantes à equipe
  - Cliente: Determina o que é mais importante, responde dúvidas dos programadores e toma decisões sobre funcionalidades.

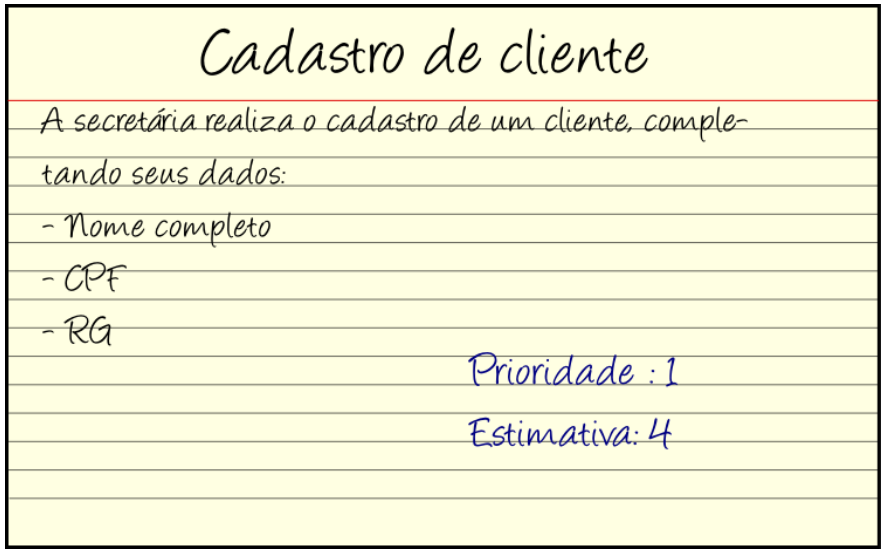
Área de trabalho informativa



AgilCoop

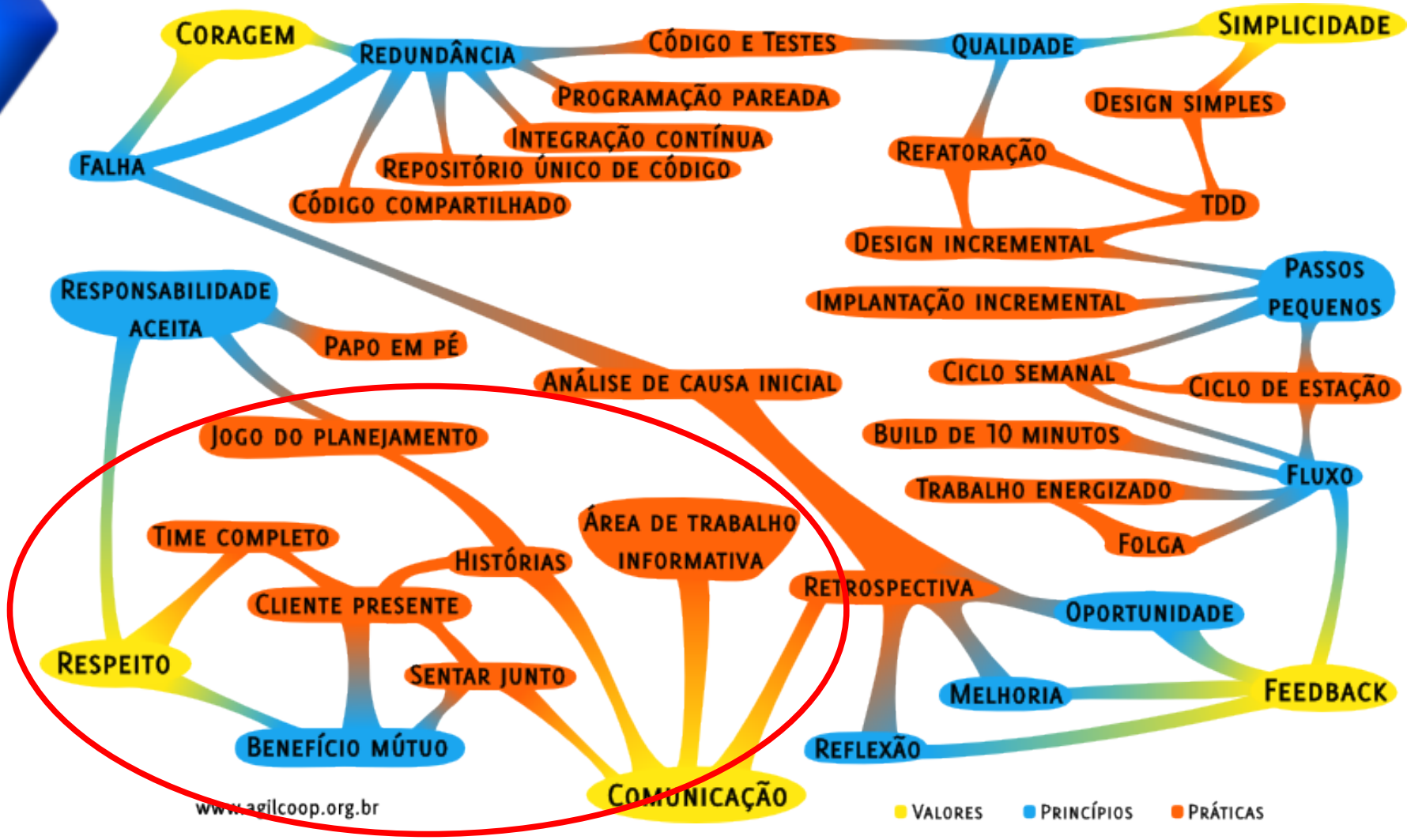
Jogo do planejamento

- Clientes escrevem e priorizam histórias
- Desenvolvedores estimam as mais prioritárias



- Suporte do *coach* e do *tracker* para evitar otimismo ou pessimismo excessivo

Localizando em XP



Sem exagerar

- Trabalho energizado:
  Balancear a intensidade do trabalho para não desgastar a equipe.

- Folga:
  Não planeje até o último minuto. Deixe uma folga para imprevistos porque eles sempre surgem.

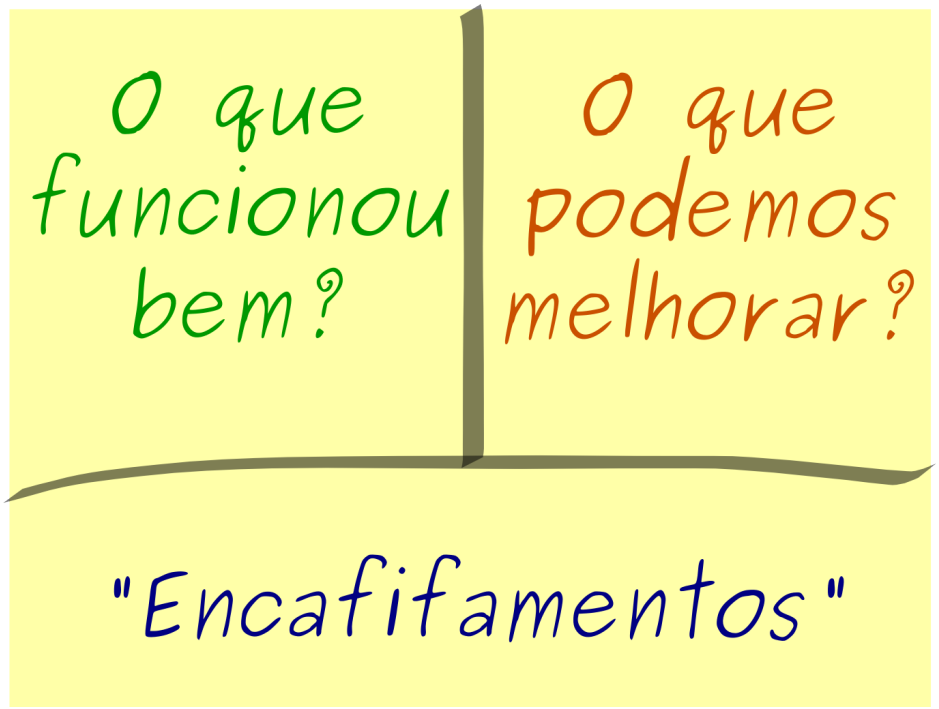
Build de 10 minutos

Fonte: http://xkcd.com/303/

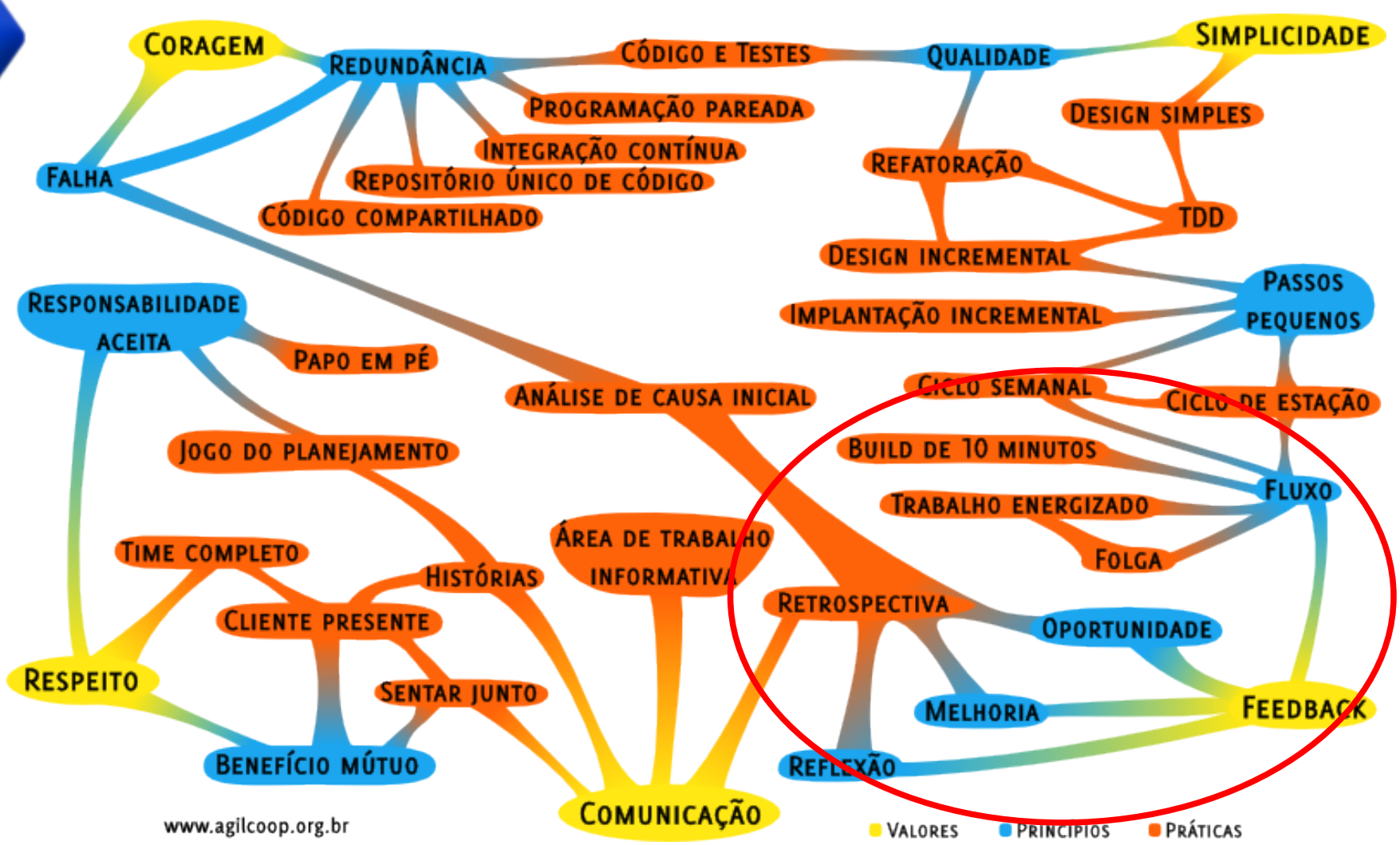
Melhorando sempre

- Análise de causa inicial

- Retrospectiva:
  Nada é perfeito.
  Tudo sempre pode melhorar e, para isso, precisa entender o que deu certo e errado.



Localizando em XP

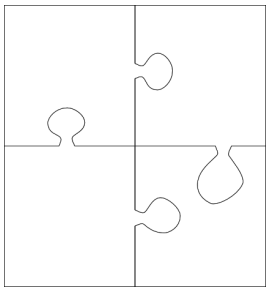


Trabalhando em equipe

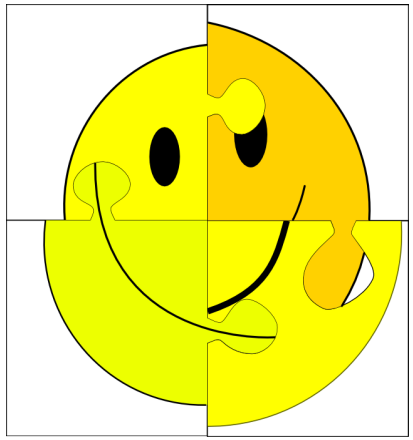
- Código compartilhado:
  Eu fiz, você arruma, nós nos ajudamos
- Padronização do estilo de código:
  Seu código e meu código devem ser quase idênticos e indistinguíveis

Código unificado

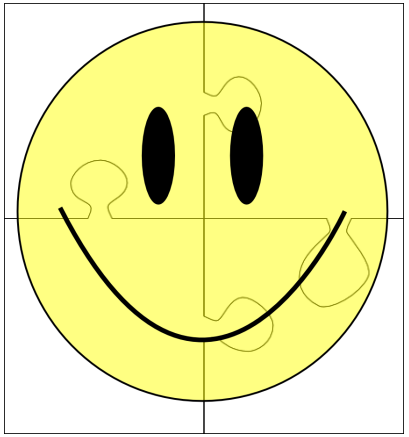
- Repositório único
- Integração contínua



SEM



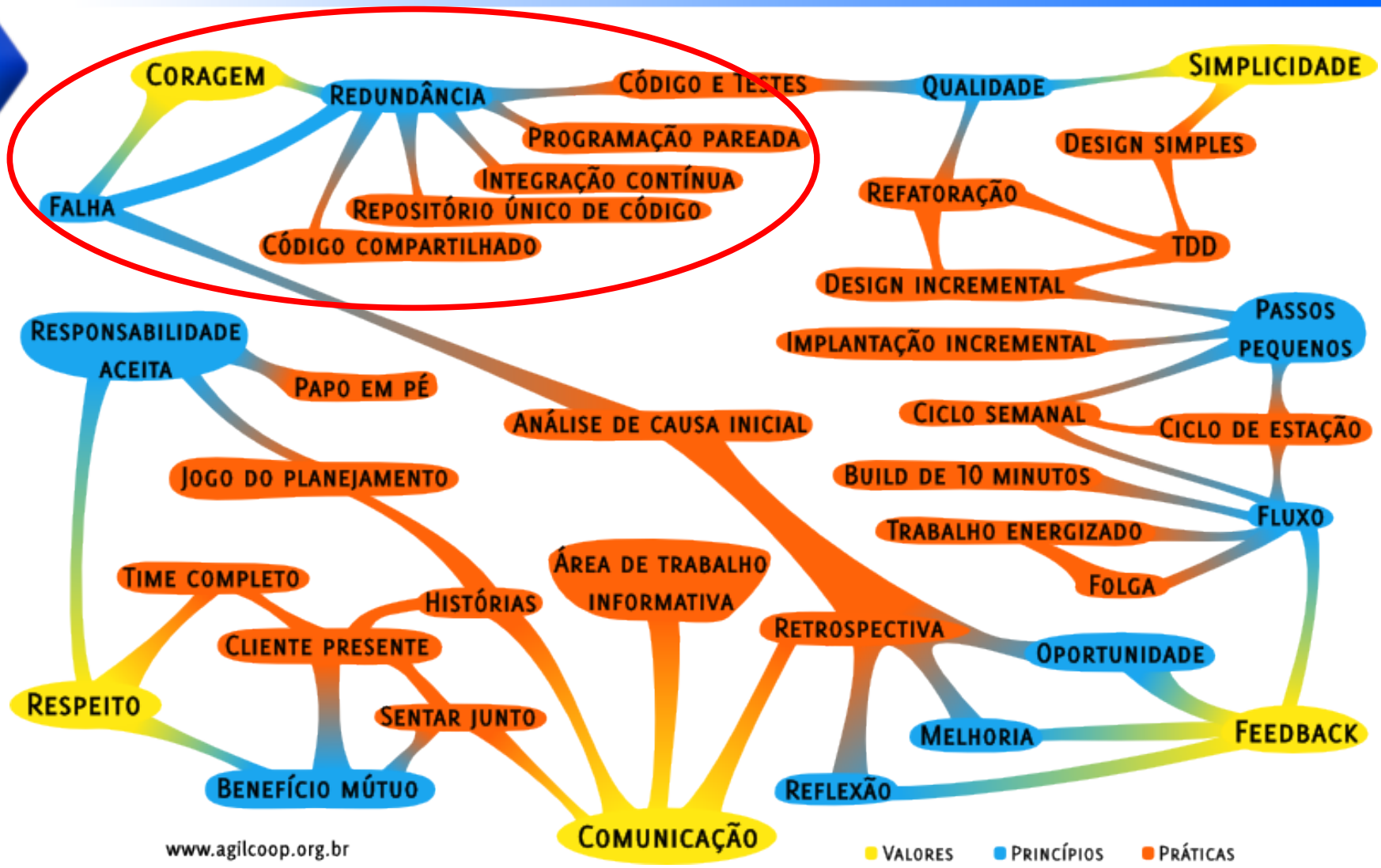
COM



Redundância

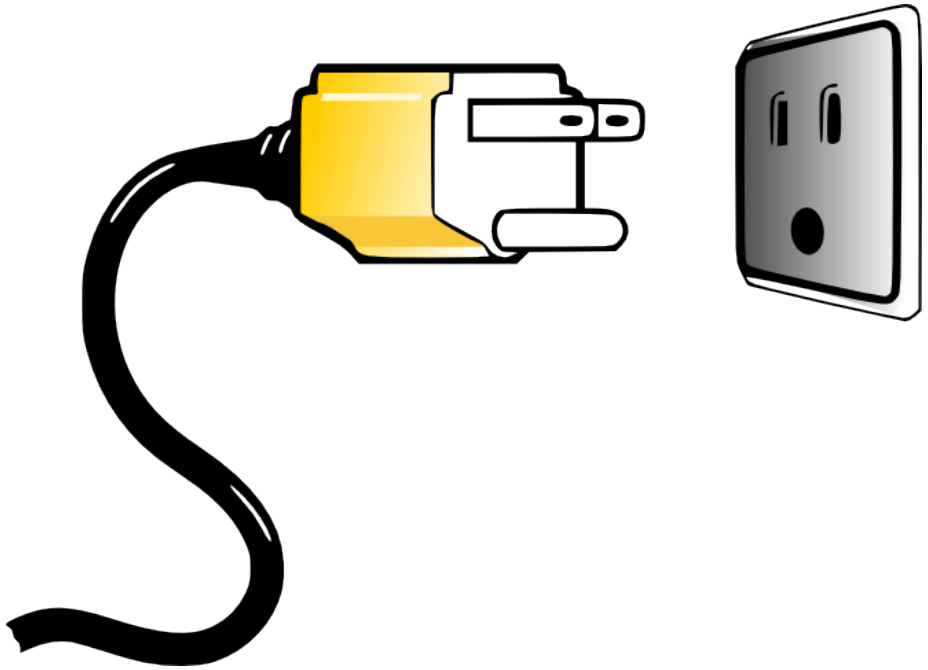
- Programação pareada:
  Distribuindo conhecimento na equipe

Localizando em XP



Previnindo defeitos

- Auto-inspeção (*mistake proof*)



Testes são a especificação!

Testes Manuais?

Geralmente são:

1) Escolher alguns valores
2) Executar o programa
3) Visualizar o resultado

Problemas:

- Demorado
- Cansativo
- Poucos casos
- Casos simples
- Implementação muda
- Difícil repetir os testes
- Executado poucas vezes
- Ninguém quer testar

Testes Automatizados!

- Código que testa outro código e verifica o resultado
- Um teste que pode ser executado facilmente várias vezes
- Testar não é depurar
  - Testar é buscar erros
  - Depurar é seguir o fluxo para identificar um erro conhecido
- “Testes podem mostrar a presença de erros, não a sua ausência” (Dijkstra)

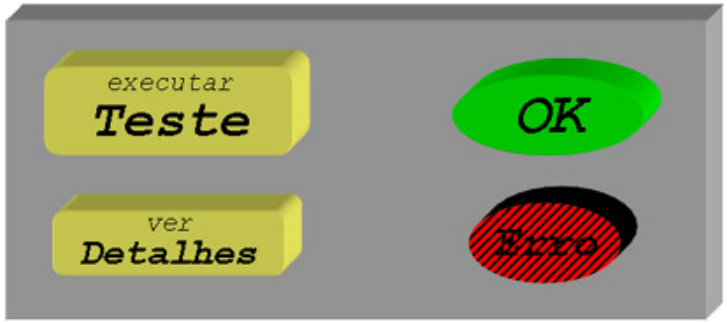
Porque testar?

- Ajuda a encontrar erros mais cedo
- Novas funcionalidades podem quebrar as funcionalidades antigas
- Dá suporte para mudanças de configuração no ambiente
- Melhora a qualidade do código, diminuindo o número de erros

Como e o quê?

A execução deve ser ágil e rápida

- Com ferramentas, scripts ou suite de testes
- Resultado deve ser simples
  - OK ou
  - Lista de Erros



Tipos de testes:

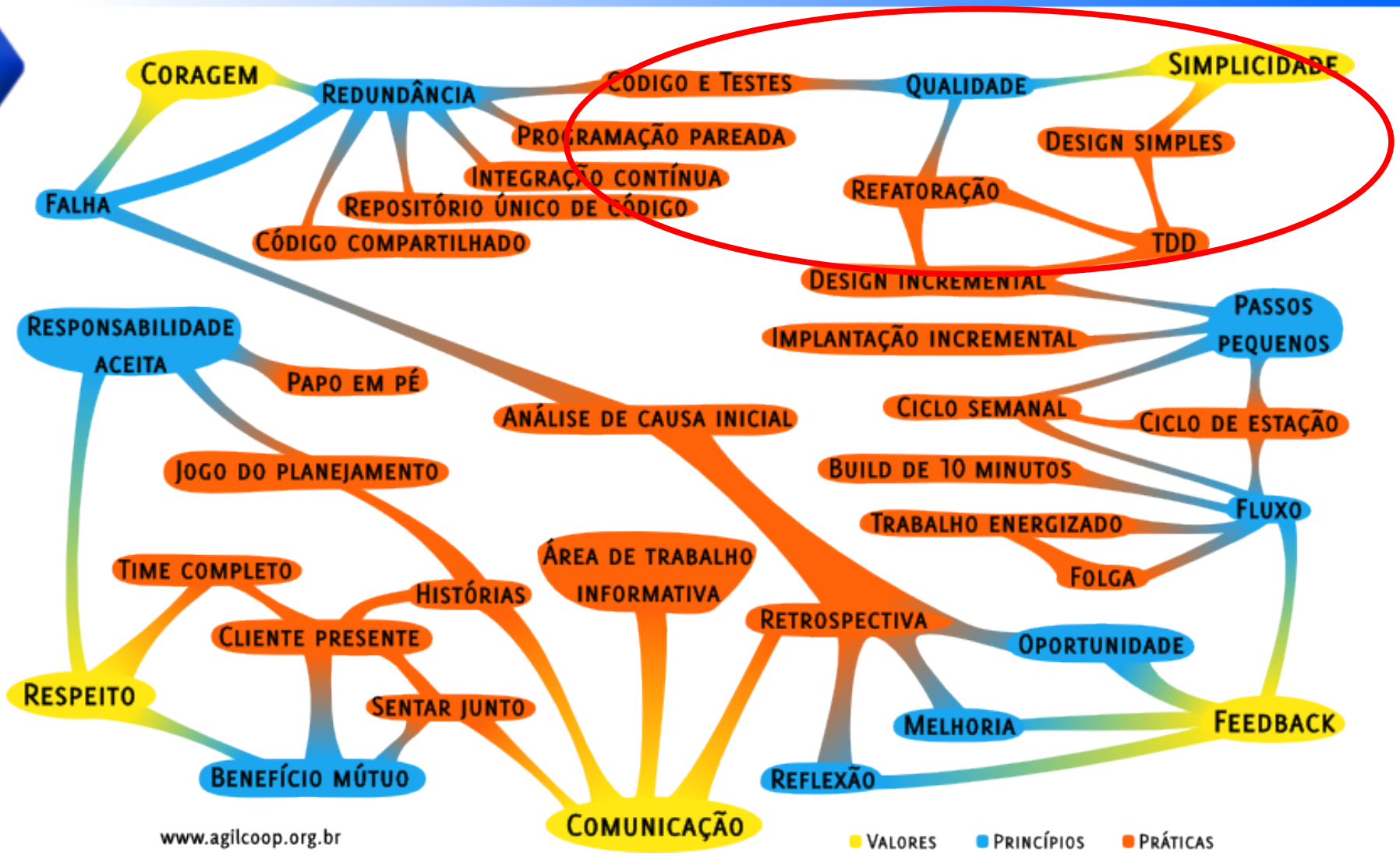
- Unidade
- Integração
- Aceitação
- Interface
- Desempenho
- Estresse
- Segurança
- …

TDD - Desenvolvimento Dirigido por Testes

Ciclo em passos pequenos:

1. Escreva um teste que falha
2. Faça o teste passar rapidamente
3. Refatore

Localizando em XP



Design Incremental

- Simples
  YAGNI: You Aint Gonna Need It
  “Você não vai precisar disso”

- Refatoração
  DRY: Dont Repeat Yourself
  “Não se repita”

Constante manutenção

- Evitando débito técnico:
  - Refatoração e Design Simples

Complexidade é o inimigo invisível!

Refatoração

- Uma [pequena] modificação no sistema que não altera o seu comportamento funcional,
- mas que torna o código mais fácil de ser entendido e menos custoso de ser alterado:
  - simplicidade
  - flexibilidade
  - clareza
- O desempenho pode piorar, otimizar não é o objetivo da refatoração

O espírito da Refatoração

Antes

O espírito da Refatoração

Antes

(refatoração) →

O espírito da Refatoração

Antes

(refatoração)

Depois

Exemplos de refatoração

- Alterar nomes de variáveis, métodos e objetos para melhorar legibilidade do código
  - String calc_imp_rnd_str(int a, Txs tx, int inc) -> String calculaImpostoDeRenda(int ano, Imposto taxas, int rendimento)
- Mudar parâmetros de métodos para que fiquem mais claros
- Melhorar o design da arquitetura
- Eliminar duplicação de código
- Flexibilizar métodos para novos usos

Refatoração sempre existiu...

- Visível na comunidade Smalltalk nos anos 80/90
- Ralph Johnson (academia)
- Kent Beck (indústria)

- Melhorar o código sempre foi uma prática implícita
- Com sua popularização (e aceitação acadêmica) surge um vocabulário de refatorações comuns e um catálogo compartilhado
- Assim podemos utilizá-las mais sistematicamente.
- Podemos aprender novas técnicas, ensinar uns aos outros.

Sintomas para refatorar

- Quando você encontra código antigo que não entende de primeira facilmente
- Ao ler código feito por outros programadores, e perceber que ele não está claro (durante revisões por exemplo)
- Quando precisa consertar uma falha, ou adicionar funcionalidade
- Regra dos 3: *Once and Only Once*

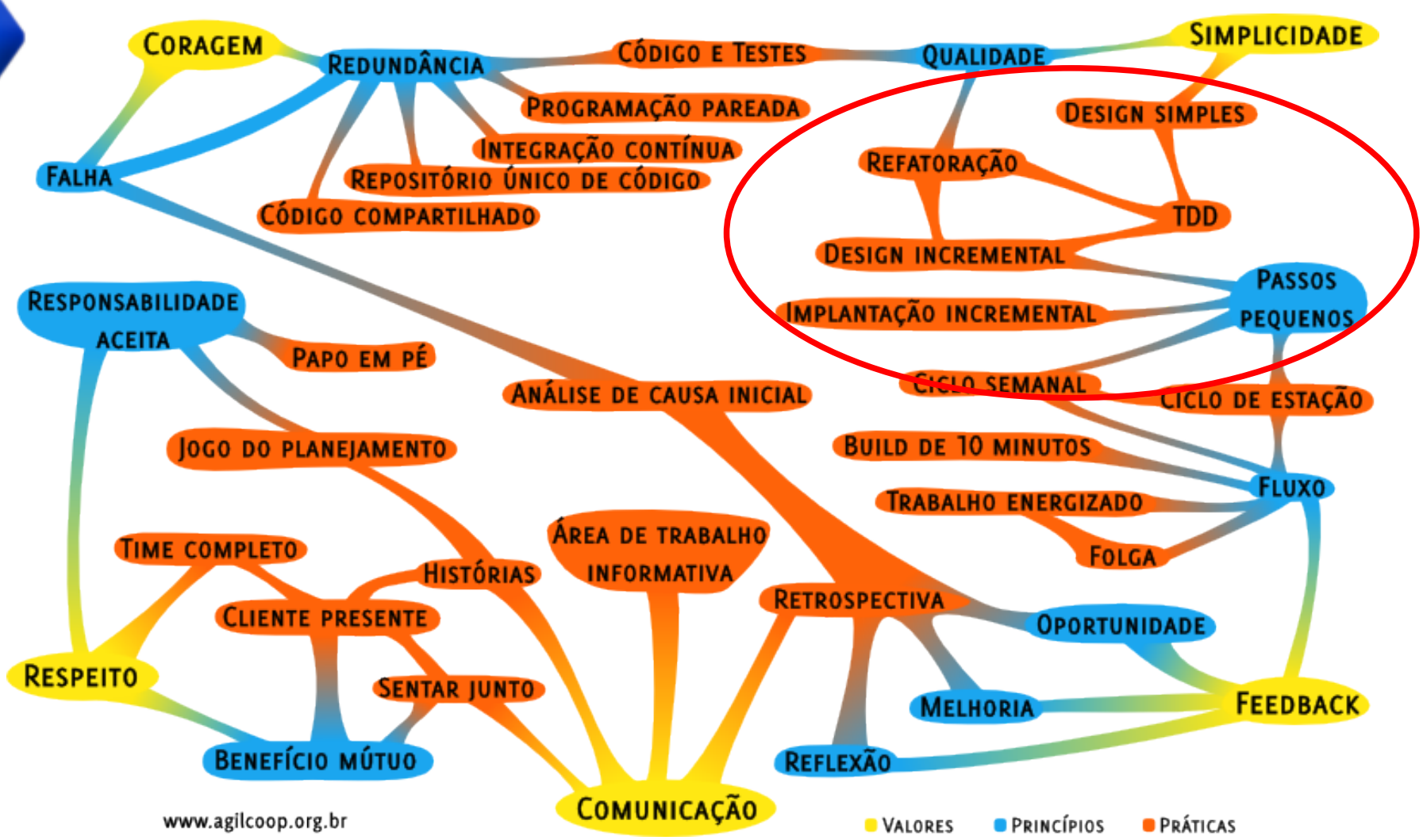
Em cada refatoração

- Antes de começar a refatoração, verifique se você tem um conjunto sólido de testes para verificar a funcionalidade do código a ser refatorado.
- Refatorações podem adicionar erros.
  - porém, como são feitas em pequenos passos, é fácil recuperar-se de uma falha
- Os testes vão ajudá-lo a detectar erros se eles forem criados.

Como fazer?

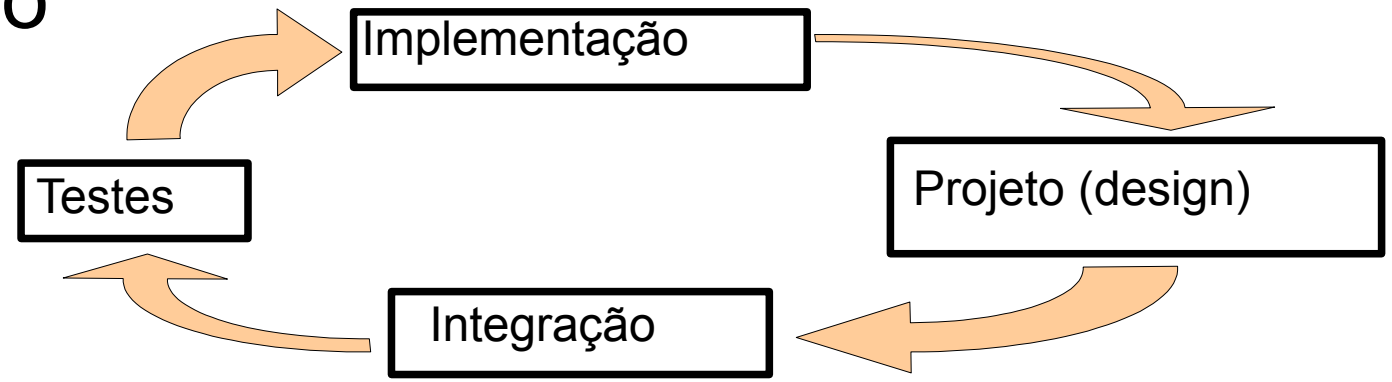
- Em geral, um *passo de refatoração* é tão simples que parece que ele não vai ajudar muito.
  - Cada passo é trivial.
  - Demora alguns segundos ou alguns poucos minutos para ser realizado.
  - É uma operação sistemática e óbvia.
- Mas quando se juntam 50 passos, bem escolhidos, em seqüência, o código melhora radicalmente.

Voltando para XP



Um dia de XP

- Escolhe uma história do cliente.
- Procura um par livre e um computador.
- Seleciona um “cartão de história”.
- Discute modificações recentes no sistema
- Discute história do cliente
- Entra no ciclo:



Dúvidas?

agilcoop@agilcoop.org.br

?

www.agilcoop.org.br

(artigos + agilcast)