

The background features abstract, overlapping green geometric shapes, primarily triangles and polygons, in various shades of green, creating a modern and dynamic visual effect.

# Princípios e Metodologias Ágeis

Marco Antonio Colombo da Silva

# A Aliança Ágil



- ▶ A dificuldade na entrega de softwares e insatisfação dos clientes levavam as grandes empresas nos anos 2000 a acreditarem que seus processos não eram eficientes e a aumentar ainda mais o número de passos e complexidade de seus processos.
- ▶ Um grupo de especialistas no setor autodenominados Aliança Ágil, se reuniu em 2001 e criaram o Manifesto da Aliança Ágil.



# Valores

- ▶ Valorizar mais:
- ▶ **Indivíduos e interações** do que processos e ferramentas.
- ▶ **Software em funcionamento** do que documentação abrangente.
- ▶ **Colaboração com o cliente** do que negociação de contratos.
- ▶ **Resposta a mudanças** do que seguir um plano.
  
- ▶ Mesmo havendo valor nos itens à direita, valorizamos mais os itens à esquerda.



# Indivíduos e interações mais que processos e ferramentas

- ▶ As pessoas são o ingrediente mais importante do sucesso.
- ▶ Trabalhar bem com outras pessoas é mais importante que o talento em programação.
- ▶ Ferramentas corretas são importantes, contudo, excesso de ferramentas enormes e difíceis é tão ruim quando a falta delas.
- ▶ Trabalhe para criar a equipe e depois a deixe configurar o ambiente de acordo com a necessidade.



# Software em funcionamento mais que documentação abrangente

- ▶ Software sem documentação é um desastre.
- ▶ Documentação demais é pior que pouca documentação.
- ▶ Produzir documentação de fácil leitura para seres humanos.
- ▶ Demoram para serem produzidos e para se manterem sincronizados com o código.
- ▶ Escrever documento breve do fundamento lógico e da estrutura.
- ▶ Transferir conhecimentos detalhados sobre o sistema por meio da interação humana.

# Colaboração com o cliente mais que negociação de contratos

- ▶ Software não pode ser pedido como uma mercadoria (cronograma fixo e preço fixo).
- ▶ *Feedback* regular e frequente do cliente.
- ▶ Um bom contrato não especifica detalhadamente um software e sim o a forma de trabalho e interação com o cliente.

# Respostas a mudanças mais que seguir um plano

- ▶ Capacidade para responder a mudanças é determinante para o sucesso.
- ▶ Planos devem ser flexíveis e se adaptar a mudanças do negócio e de tecnologia.
- ▶ A estrutura de um gráfico de Gantt se degrada à medida que a equipe obtém conhecimento sobre o projeto e o cliente obtém conhecimento sobre as necessidades da equipe.
- ▶ O plano muda em forma e não apenas em datas.
- ▶ Fazer planos detalhados para a semana seguinte, planos aproximados para os próximos meses e planos extremamente rudimentares para o além disso.

# Princípios

- Os valores mostrados anteriormente inspiraram 12 princípios que diferenciam um conjunto de práticas ágeis de um processo peso-pesado.





# Princípio 1

Nossa maior prioridade é satisfazer o cliente com a entrega antecipada e contínua de software de valor.

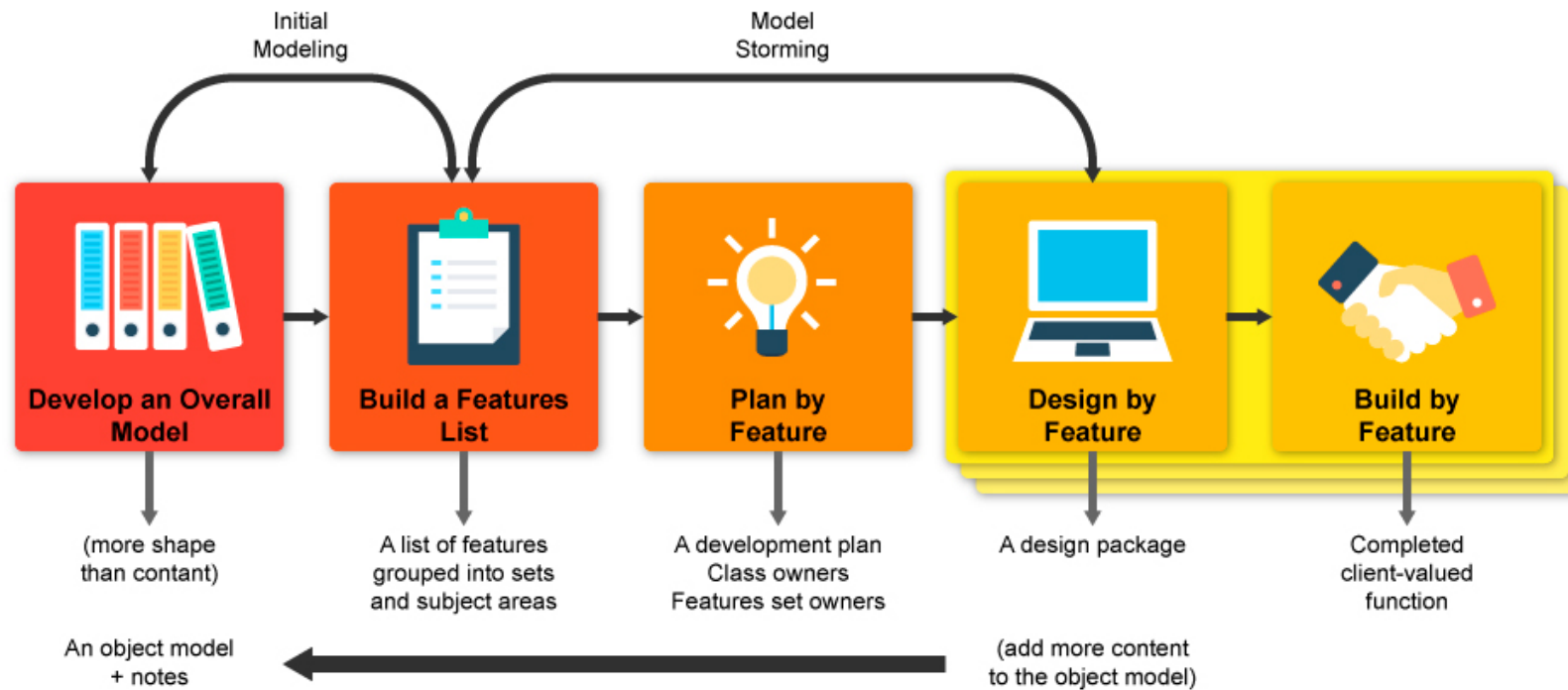
- Um artigo do MIT revelou que a entrega antecipada de software com funcionamento parcial tem forte correlação com entrega final de qualidade.  
(Martin 2011)



# Princípio 1

- ▶ **FDD - Feature Driven Development**
- ▶ Entrega de resultados frequentes, tangíveis e funcionais
- ▶ Durante a fase de implementação, a FDD sugere que sempre que alguma funcionalidade for implementada e testada ela deve ser disponibilizada aos usuários.

## ► FDD - Feature Driven Development



# Princípio 2

Mudanças nos requisitos são bem-vindas mesmo com o desenvolvimento já adiantado.

- Uma equipe ágil trabalha arduamente para manter a estrutura do software seja flexível, a fim de que, quando os requisitos mudarem, o impacto no sistema seja mínimo.



# Princípio 3

Entregar com frequência software funcionando, de poucas em poucos meses, dando preferência à escala de tempo mais curta.

- Entrega de um punhado de documentos ou planos não são verdadeiras entregas.



# Princípio 4

Executivos e desenvolvedores devem trabalhar juntos diariamente ao longo de todo o projeto.

- ▶ A interação deve ser significativa e frequente. Um projeto de software deve ser continuamente orientado.



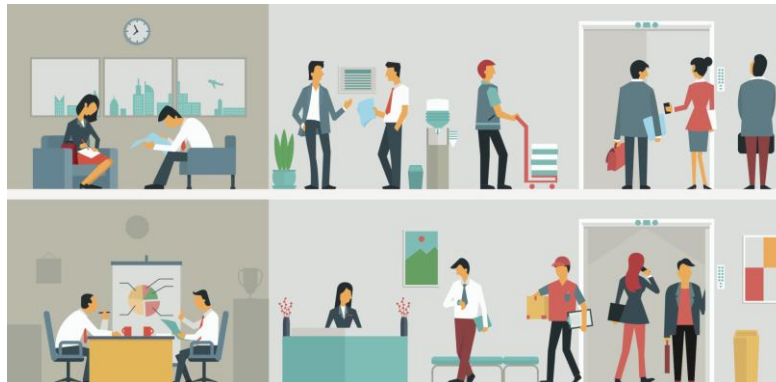
# Princípio 4

- ▶ **Dynamic Systems Development (DSDM)**
- ▶ **Papéis para Stakeholders**
  - ▶ Usuário embaixador
  - ▶ Usuário visionário
  - ▶ Usuário conselheiro
  - ▶ Patrocinador executivo

# Princípio 5

Construir projetos com indivíduos motivados. Fornecer a eles o ambiente e o apoio necessários e ter confiança de que eles farão o trabalho.

- As pessoas são o mais importante e processo, ambiente, administração etc devem mudar caso estejam prejudicando as pessoas.





# Princípio 5

- ▶ **SCRUM**

- ▶ São responsabilidades da equipe:

- ▶ Fazer as estimativas necessárias;
- ▶ Definir as tarefas que serão realizadas;
- ▶ Apresentar o produto ao cliente;

# Princípio 6

O método mais eficiente e eficaz de transmitir informações para dentro de uma equipe de desenvolvimento é a conversa face-a-face.

- Os documentos escritos são criados e incrementados no mesmo cronograma que o software e somente quando necessário.



# Princípio 6

- ▶ **Extreme Programming (XP)**
- ▶ A XP **exige** a que todo o código seja implementado em dupla e no mesmo computador (pair programming).
  - ▶ Problema dividido,
  - ▶ Troca de experiência,
  - ▶ na ausência de uma delas a outra pode prosseguir.
- ▶ Stand up meeting

# Princípio 6

- ▶ **SCRUM**

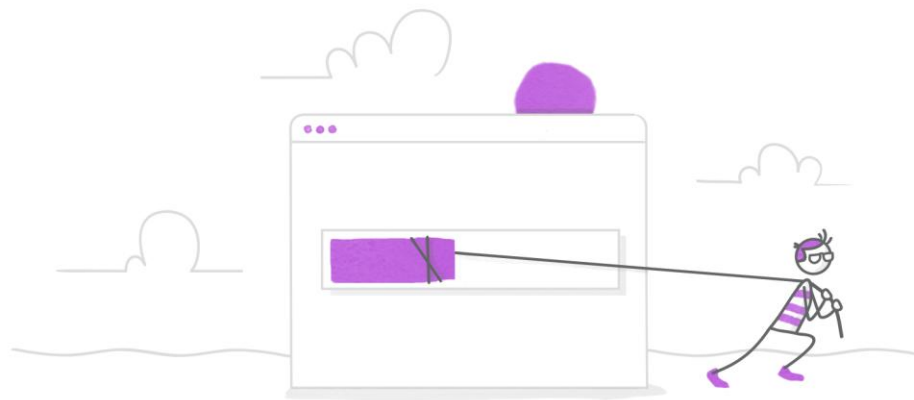
- ▶ **Daily SCRUM**

- ▶ máximo 15 minutos
- ▶ de preferência em pé
- ▶ de preferência após o almoço

# Princípio 7

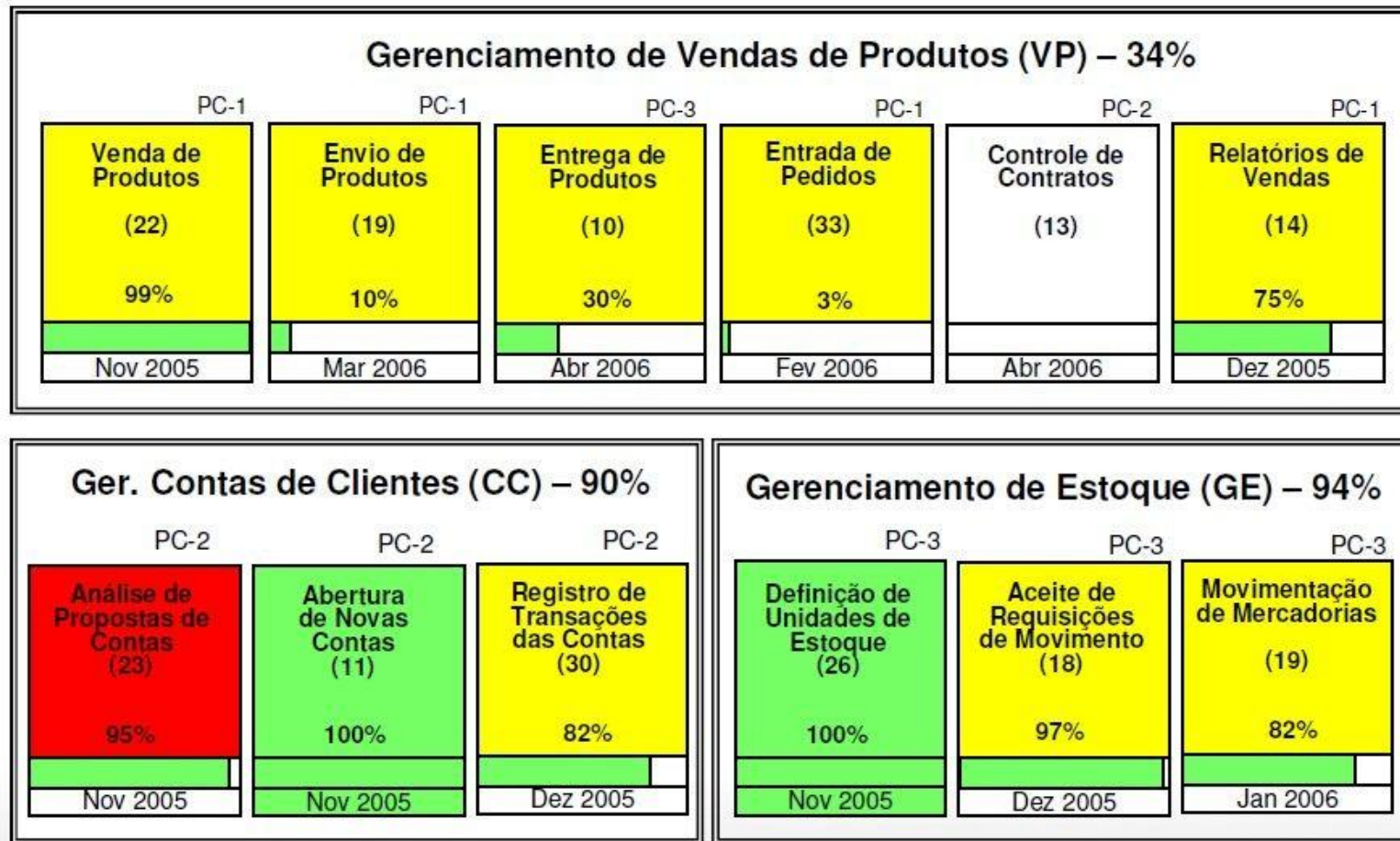
Software funcionando é a principal medida do progresso.

- ▶ Não medir progresso em termos de fase, nem pelo volume de documentação ou pela quantidade de código.



# Princípio 7

## ► FDD - Feature Driven - Exemplo de Relatório de Progresso



Legenda:   Em andamento   Atenção   Completada   Barra de Progresso   Não iniciada

# Princípio 8

Processos ágeis promovem o desenvolvimento sustentável.

- ▶ As equipes não devem gastar a energia de amanhã para fazer um pouco mais hoje. Elas trabalham em uma velocidade que lhes permite manter os padrões de qualidade mais alta durante todo o projeto.



# Princípio 8

- ▶ Extreme Programming (XP)
- ▶ A XP **proíbe** que a equipe de projeto trabalhe fora de seu horário normal.
- ▶ O ritmo sustentável sugerido pela XP é 40 horas semanais.

(Sbrocco, 2012)



# Princípio 9

Atenção contínua à excelência técnica e ao bom projeto aumenta a agilidade.

- ▶ O código deve ser limpo e robusto no momento em que é produzido.
- ▶ Sem “Depois eu arrumo”.



# Princípio 10

Simplicidade - a arte de maximizar o volume de trabalho que não precisou ser feito

- ▶ Tomar o caminho mais simples e coerente com o objetivo.
- ▶ Não antecipar problemas futuros.
- ▶ Amanhã será mais fácil alterar o que foi feito com simplicidade e qualidade hoje.



# Princípio 11

As melhores arquiteturas, requisitos e projetos emergem de equipes auto-organizadas.

- ▶ Responsabilidades não são dadas de fora aos membros da equipe.
- ▶ Nenhum membro é responsável sozinho pela arquitetura, pelos requisitos ou pelos testes.



# Princípio 11

- ▶ **Dynamic Systems Development (DSDM)**
  - ▶ **Equipes com poder de decisão**
    - ▶ as pessoas envolvidas no desenvolvimento devem ter conhecimento e autonomia suficiente para decidir o destino do sistema;
    - ▶ não é tolerável aguardar decisões por um longo período de tempo.
- (Sbrocco 2012)

# Princípio 11

- ▶ **SCRUM**
- ▶ São responsabilidades do SCRUM Master:
  - ▶ Remover impedimentos;
  - ▶ Atuar como escudo para interferências externas.

# Princípio 12

Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz e, então adapta e ajusta seu comportamento de forma correspondente.

- ▶ Ajustar continuamente sua organização, suas regras, suas convenções, seus relacionamentos etc.



# Artigo para Análise Crítica

- **Agile methods for embedded systems development - a literature review and a mapping study**

Kaisti M, Rantala V, Mujunen T, Hyrynsalmi S, Könnölä K, Mäkilä T, Lehtonen T

**EURASIP Journal on Embedded Systems**

2013 vol: 2013 (1) pp: 15

- DOI 10.1186/1687-3963-2013-15

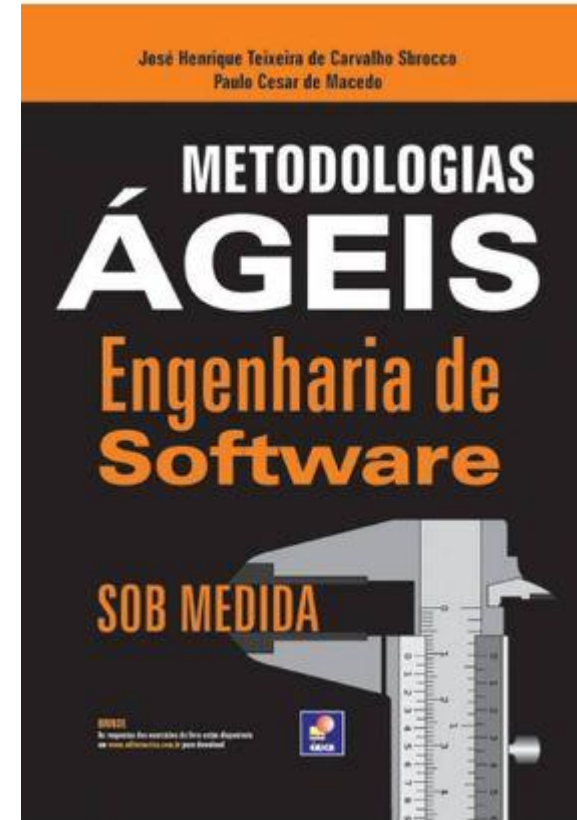
# Fontes

**Metodologias Ágeis - Engenharia de Software**

José Henrique Teixeira de Carvalho Sbrocco

Paulo Cesar de Macedo

2012





# Fontes

Princípios, Padrões e Práticas Ágeis em C#

Robert C. Martin

Micah Martin

2011

