Robert Cecil martin (uncle bob), está por trás do manifesto ágil

- Um dos gargalos principais no des. de um software esta na manutenção, um código que funciona mas está mal escrito gera prejuízos enormes

- 10x tentando entender o código existentes do que escrevendo novos códigos

- Deve ter uma boa arquitetura

- Torna o sistema preparada para novas atualizações e melhorias

- Reduz gastos desnecessários para manutenção

**1 – Definição dos nomes (variáveis, funções, parâmetro)**

- O nome precisa passar uma ideia central

- Não tenha medo de nomes grandes

- Métodos e funções como VERBOS

- Classes e Objetos SUBSTANTIVOS (classe caneta, método escrever)

**2 – Regra do escoteiro**

Deixe seu código mais limpo de como estava antes de vc mexer

**3 – Você é o autor do código**

- Funções pequenas e devem ser ainda menores

- Funções devem executar apenas uma tarefa

**4 – DRY (Don’t repeat Yourself)**

Não pode existir 2 partes de um sistema que realiza a mesma função.

**5 – Comentários**

Depois que você edita o código dificilmente o kra altera o código.

**6 – Tratamentos de erros**

**Teste FIRST**

**- Fast:** testes rápidos para serem validados a todo momento

**- Indeependent:** independência, para que um teste não cause um efeito cascata quando há uma falha o que dificulta varias analises

**- Repeatable:** repetir diversas vezes em ambientes diferentes SELF-- **Validation:** os testes bens escritos retornam respostas true ou false para não ser subjetivo  
**- Timely:** pontualidade (cada teste antes do código)