

The  
Pragmatic  
Programmers

## Test-Driven Development for Embedded C

James W. Grenning

Forewords by Jack Ganssle  
and Robert C. Martin

*Edited by Jacquielyn Carter*

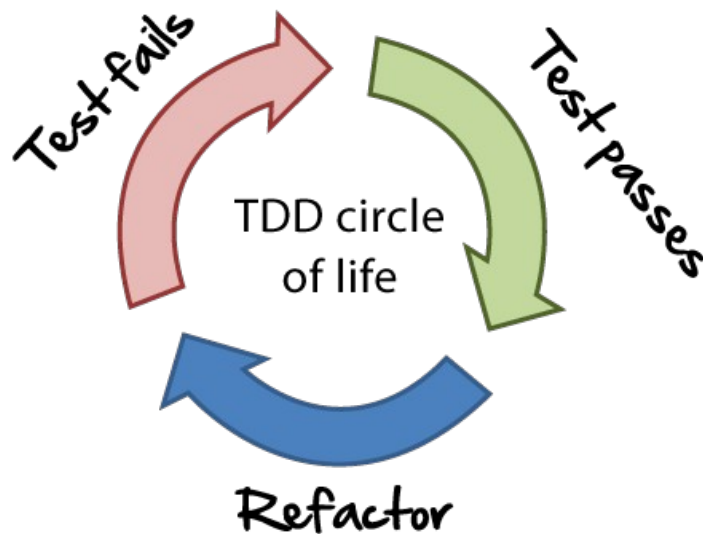


# O que é Test Driven Development (TDD)?

Autor: Rodolfo Dirack

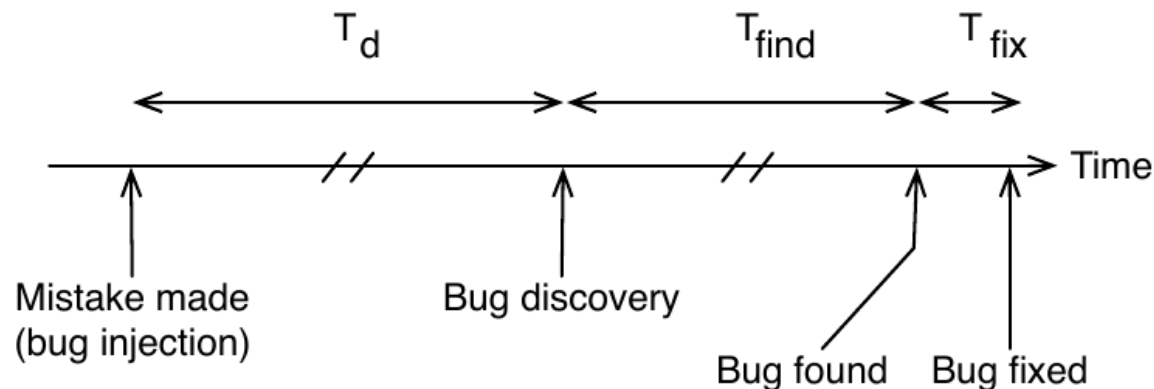
# Oque é TDD?

- TDD (Test Driven Development – Desenvolvimento orientado a testes) é uma técnica incremental de desenvolvimento de software
- Funciona em pequenos ciclos chamados de *baby steps*



# Como é sem TDD?

- Fazer todo o programa e depois a depuração
- Cada erro trás outros erros
- Depuração difícil

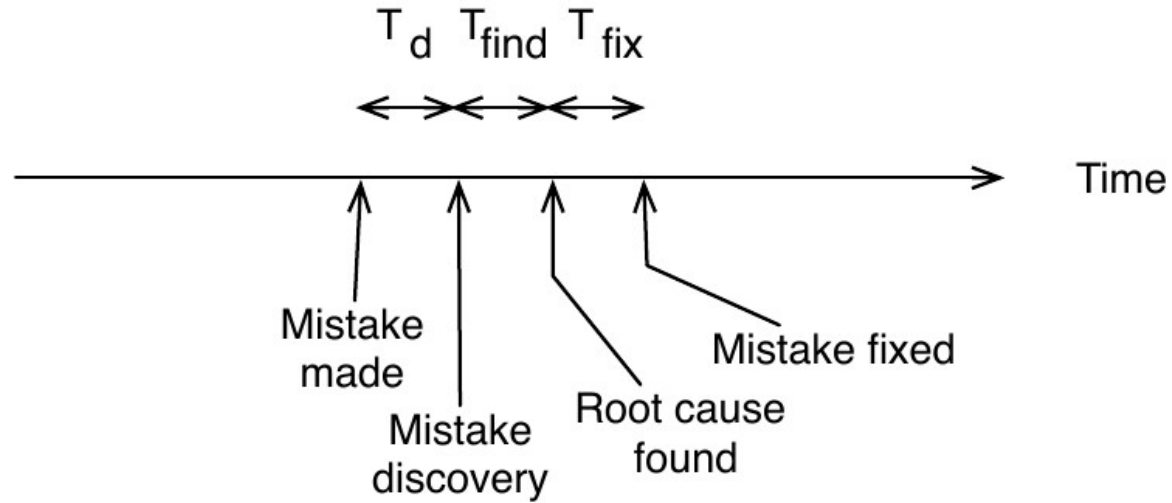




# Como é com TDD?

- Nenhum código de produção é feito sem escrever um teste antes
- Testes são pequenos
- Testes são automatizados
- Expressar o comportamento desejado do código no teste
- TDD NÃO é uma técnica de testes!
- TDD é divertido!

# Como é com TDD?

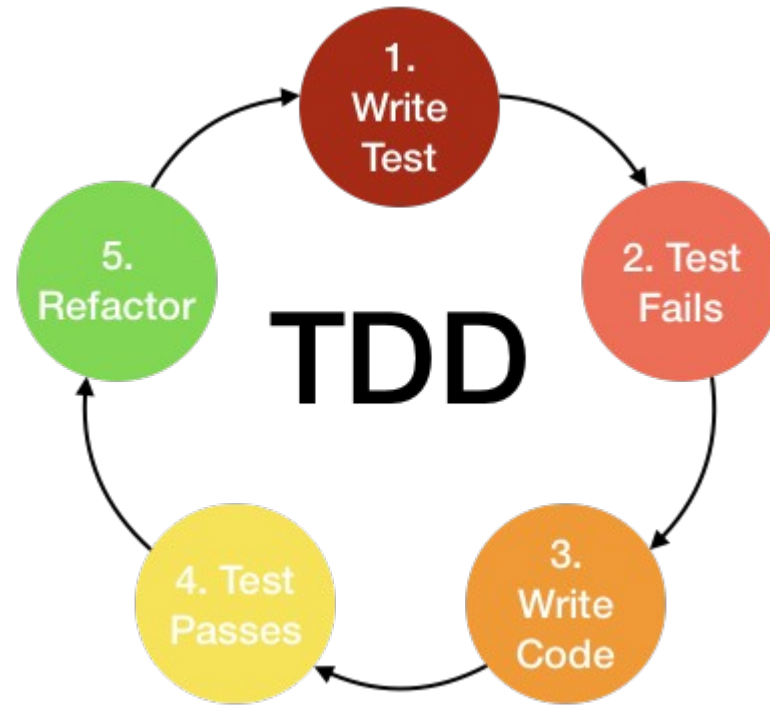




# Passos do TDD

- Adicionar um pequeno teste
- Rodar todos os testes (Teste falha)
- Fazer pequenas mudanças no código para o teste passar
- Rodar todos os testes (Teste passou)
- Refatorar o código (Remover código duplicado, etc)

# Passos do TDD





# Benefícios do TDD

- Menos bugs
- Menos tempo de depuração
- Menos efeitos colaterais
- Testes são documentação! Trazem confiança!
- Testes monitoram o progresso
- TDD é divertido e recompensador!





# O que é Refatoração?

- Alteração feita no código de um software para torná-lo:
  - Mais fácil de compreender;
  - Mais fácil de modificar;
  - Sem alterar o seu comportamento observável;



# Por que refatorar?

- Para obter um software de qualidade!
- Código limpo e claro:
  - ➔ Fácil de ler (expressar intenção)
  - ➔ Fácil de compreender
  - ➔ Fácil de promover mudanças

# Code Refactoring (Martin Fowler)



“Uma série de passos pequenos, cada um dos quais muda a estrutura interna do programa sem alterar o seu comportamento externo.”