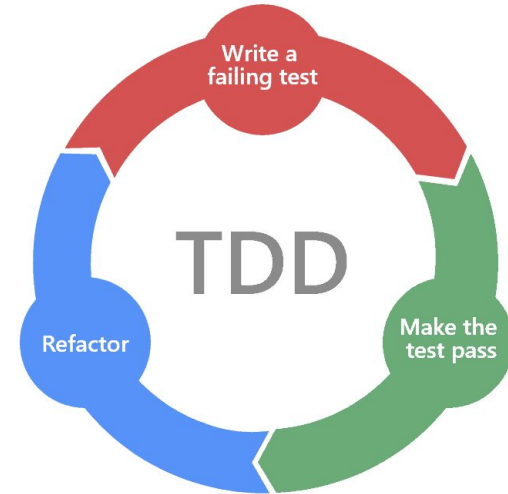


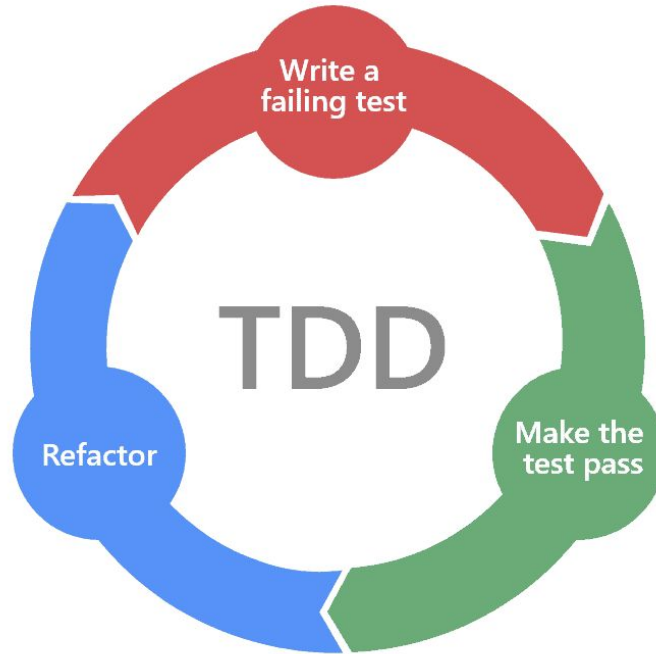
# Was ist TDD

- Test Driven Development (TDD) ist ein Softwareentwicklungsprozess, bei dem Tests vor der eigentlichen Implementierung geschrieben werden



Quelle: <https://marsner.com/>

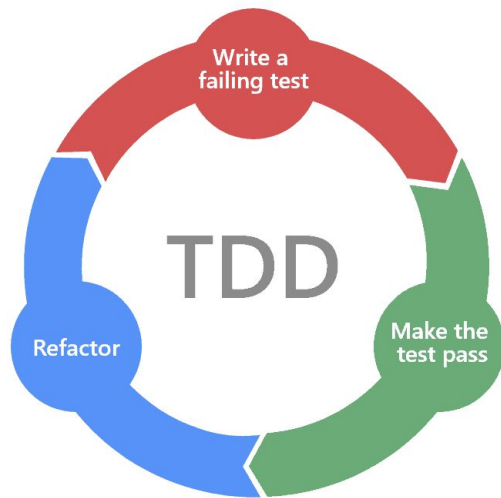
# Test Driven Design



Quelle: <https://marsner.com/>

# Grundprinzipien von TDD

1. Schreibe zuerst den Test.
2. Schreibe dann den minimalen Code, um den Test zum Bestehen zu bringen.
3. Refaktorisiere den Code, um ihn zu verbessern, ohne die Funktionalität zu ändern, und stelle sicher, dass alle Tests weiterhin bestehen.



Quelle: <https://marsner.com/>

# Vorteile von TDD

- Verbessert die Codequalität
- Frühes Erkennen von Fehlern
- Unterstützt eine agile Entwicklung
- Fördert eine bessere Testabdeckung

# Entwicklung einer Division Funktion mit TDD

## Schritt 1: Schreiben des Tests

```
// test.js
test('Division von zwei Zahlen', () => {
  expect(() => divide(6, 0)).toThrow(TypeError);
});
```

# Entwicklung einer Division Funktion mit TDD

## Schritt 2: Schreiben des minimalen Codes

```
// math.js
function divide(a, b) {
  if (b === 0) {
    throw new TypeError('Cannot divide by zero');
  }
  return a / b;
}
```

# Entwicklung einer Division Funktion mit TDD

## Schritt 3: Refaktorisierung

Keine Refaktorisierung in diesem Beispiel, da der Code bereits minimal ist und den Fehler bei einer Division durch Null behandelt.

# Zusammenfassung TDD

- TDD ist eine praktische Methode zur Entwicklung von Software.
- Es beginnt mit dem Schreiben von Tests, die das gewünschte Verhalten beschreiben.
- Dann wird der minimale Code geschrieben, um die Tests zu bestehen.
- Der Code wird anschließend refaktorisiert, um seine Qualität zu verbessern.