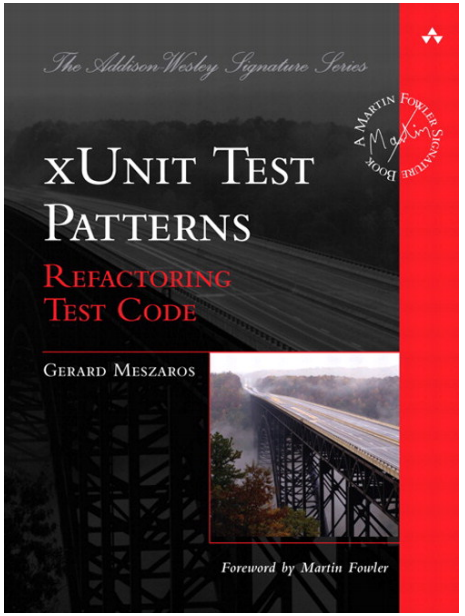
Exemplo de Teste de Software

(livro do Meszaros)



Exemplo de Teste [Meszaros]

```
public void testAddItemQuantity_severalQuantity_v1(){
  Address billingAddress = null;
  Address shippingAddress = null;
  Customer customer = null;
  Product product = null;
  Invoice invoice = null;
  try {
    // Set up fixture
    billingAddress = new Address("1222 1st St SW", "Calgary", "Alberta", "T2N
                                 2V2","Canada");
    shippingAddress = new Address("1333 1st St SW", "Calgary", "Alberta", "T2N
                                 2V2","Canada");
    customer = new Customer(99, "John", "Doe",
                               new BigDecimal("30"), billingAddress, shippingAddress);
    product = new Product(88, "SomeWidget", new BigDecimal("19.99"));
    invoice = new Invoice(customer);
    // Exercise SUT
    invoice.addItemQuantity(product, 5);
```

Exemplo de Teste

```
    // Verify outcome
    List lineItems = invoice.getLineItems();
    if (lineItems.size() == 1) {
      LineItem actItem = (LineItem) lineItems.get(0);
      assertEquals("inv", invoice, actItem.getInv());
      assertEquals("prod", product, actItem.getProd());
      assertEquals("quant", 5, actItem.getQuantity());
      assertEquals("discount", new BigDecimal("30"), actItem.getPercentDiscount());
      assertEquals("unit price", new BigDecimal("19.99"), actItem.getUnitPrice());
      assertEquals("extended", new BigDecimal("69.96"), actItem.getExtendedPrice());
    } else {
      assertTrue("Invoice should have 1 item", false);  }
  } finally {
    // Teardown
    deleteObject(invoice);
    deleteObject(product);
    deleteObject(customer);
    deleteObject(billingAddress);
    deleteObject(shippingAddress);  }  }
```

Exemplo de Teste

- O teste está longo e difícil de entender!

Exemplo de Teste

- Um problema simples de resolver: a asserção da última linha - assertTrue com um argumento falso.

```
List lineItems = invoice.getLineItems();
 if (lineItems.size() == 1) {
   LineItem actItem = (LineItem) lineItems.get(0);
   assertEquals("inv", invoice, actItem.getInv());
   assertEquals("prod", product, actItem.getProd());
   assertEquals("quant", 5, actItem.getQuantity());
   assertEquals("discount", new BigDecimal("30"),
               actItem.getPercentDiscount());
   assertEquals("unit price", new BigDecimal("19.99"),
               actItem.getUnitPrice());
   assertEquals("extended", new BigDecimal("69.96"),
               actItem.getExtendedPrice());
 } else {
   assertTrue("Invoice should have 1 item", false);  }
```

sempre vai falhar!

Exemplo de Teste

```
List lineItems = invoice.getLineItems();
 if (lineItems.size() == 1) {
   LineItem actItem = (LineItem) lineItems.get(0);
   assertEquals("inv", invoice, actItem.getInv());
   assertEquals("prod", product, actItem.getProd());
   assertEquals("quant", 5, actItem.getQuantity());
   assertEquals("discount", new BigDecimal("30"),
               actItem.getPercentDiscount());
   assertEquals("unit price",new BigDecimal("19.99"),
               actItem.getUnitPrice());
   assertEquals("extended", new BigDecimal("69.96"),
               actItem.getExtendedPrice());
 } else {
   fail("Invoice should have exactly one line item");
 }
```

Exemplo de Teste

Problema: conjunto grande de asserts.

```
List lineItems = invoice.getLineItems();
if (lineItems.size() == 1) {
  LineItem actItem = (LineItem) lineItems.get(0);
  assertEquals("inv", invoice, actItem.getInv());
  assertEquals("prod", product, actItem.getProd());
  assertEquals("quant", 5, actItem.getQuantity());
  assertEquals("discount", new BigDecimal("30"),
               actItem.getPercentDiscount());
  assertEquals("unit price",new BigDecimal("19.99"),
               actItem.getUnitPrice());
  assertEquals("extended", new BigDecimal("69.96"),
               actItem.getExtendedPrice());
} else {
  fail("Invoice should have exactly one line item");
```

Exemplo de Teste

```
List lineItems = invoice.getLineItems();
if (lineItems.size() == 1) {
  LineItem expected = new LineItem(invoice, product,5,
                      new BigDecimal("30"), new BigDecimal("69.96"));
  LineItem actItem = (LineItem) lineItems.get(0);
  assertEquals("invoice", expected, actItem);
} else {
  fail("Invoice should have exactly one line item");
}
```

Exemplo de Teste

Problema: por que tem um comando if em um teste?

```
List lineItems = invoice.getLineItems();
if (lineItems.size() == 1) {
  LineItem expected = new LineItem(invoice, product,5,
              new BigDecimal("30"), new BigDecimal("69.96"));
  LineItem actItem = (LineItem) lineItems.get(0);
  assertEquals("invoice", expected, actItem);
} else {
  fail("Invoice should have exactly one line item");
}
```

Exemplo de Teste

```
List lineItems = invoice.getLineItems();
assertEquals("number of items", 1,lineItems.size());
LineItem expected = new LineItem(invoice, product,5,
            new BigDecimal("30"), new BigDecimal("69.96"));
LineItem actItem = (LineItem) lineItems.get(0);
assertEquals("invoice", expected, actItem);
```

Exemplo de Teste

- Esta asserção não poderia ser mais óbvia?

```
LineItem expected = new LineItem(invoice, product,5,
                new BigDecimal("30"), new BigDecimal("69.96"));
assertContainsExactlyOneLineItem(invoice, expected);
```

método definindo uma
asserção customizada

Versão Atual do Teste (1)

```
public void testAddItemQuantity_severalQuantity_v1(){
  Address billingAddress = null;
  Address shippingAddress = null;
  Customer customer = null;
  Product product = null;
  Invoice invoice = null;
  try {
    // Set up fixture
    billingAddress = new Address("1222 1st St SW", "Calgary", "Alberta",
                                 "T2N 2V2","Canada");
    shippingAddress = new Address("1333 1st St SW", "Calgary", "Alberta",
                                  "T2N 2V2","Canada");
    customer = new Customer(99, "John", "Doe", new BigDecimal("30"),
                                  billingAddress, shippingAddress);
    product = new Product(88, "SomeWidget", new BigDecimal("19.99"));
    invoice = new Invoice(customer);
```

Versão Atual do Teste (1)

```
  // Exercise SUT
    invoice.addItemQuantity(product, 5);
  // Verify outcome
    LineItem expected = new LineItem(invoice, product,5,
                  new BigDecimal("30"), new BigDecimal("69.96"));
    assertContainsExactlyOneLineItem(invoice, expected);
} finally {
    // Teardown
    deleteObject(invoice);
    deleteObject(product);
    deleteObject(customer);
    deleteObject(billingAddress);
    deleteObject(shippingAddress);  }  }
```

Exemplo de Teste

- O que o código final do teste faz?

```
} finally {
    // Teardown
    deleteObject(invoice);
    deleteObject(product);
    deleteObject(customer);
    deleteObject(billingAddress);
    deleteObject(shippingAddress);  }  }
```

Exemplo de Teste - Meszaros

- Os objetos podem ser registrados pelo framework e depois removidos. O registro consiste em colocar o objeto em uma coleção.

```
List testObjects;
protected void setUp() throws Exception {
  super.setUp();
  testObjects = new ArrayList();
}
protected void registerTestObject(Object testObject) {
  testObjects.add(testObject);
}
```

Exemplo de Teste - Meszaros

```
public void tearDown() {
  Iterator i = testObjects.iterator();
  while (i.hasNext()) {
  try {
     deleteObject(i.next());
  } catch (RuntimeException e) {
      }
  }
}
```

Versão Atual do Teste (2)

```
public void testAddItemQuantity_severalQuantity_v8(){
  Address billingAddress = null;
  Address shippingAddress = null;
  Customer customer = null;
  Product product = null;
  Invoice invoice = null;
  // Set up fixture
  billingAddress = new Address("1222 1st St SW", "Calgary", "Alberta",
                               "T2N 2V2", "Canada");
  registerTestObject(billingAddress);
  shippingAddress = new Address("1333 1st St SW", "Calgary", "Alberta",
                               "T2N 2V2", "Canada");
  registerTestObject(shippingAddress);
  customer = new Customer(99, "John", "Doe", new BigDecimal("30"),
                               billingAddress, shippingAddress);
  registerTestObject(shippingAddress);
```

Versão Atual do Teste (2)

```
    product = new Product(88, "SomeWidget", new BigDecimal("19.99"));
    registerTestObject(shippingAddress);
    invoice = new Invoice(customer);
    registerTestObject(shippingAddress);

    // Exercise SUT
    invoice.addItemQuantity(product, 5);

    // Verify outcome
    LineItem expected = new LineItem(invoice, product, 5,
                    new BigDecimal("30"), new BigDecimal("69.96"));
    assertContainsExactlyOneLineItem(invoice, expected);
}
```

Exemplo de Teste

- Para que declarar as variáveis e inicializá-las com null? Antes era necessário por causa do bloco finallly.

Versão Atual do Teste (3)

```
public void testAddItemQuantity_severalQuantity_v9(){
  // Set up fixture
  Address billingAddress = new Address("1222 1st St SW", "Calgary",
                "Alberta", "T2N 2V2", "Canada");
  registerTestObject(billingAddress);
  Address shippingAddress = new Address("1333 1st St SW", "Calgary",
                "Alberta", "T2N 2V2", "Canada");
  registerTestObject(shippingAddress);
  Customer customer = new Customer(99, "John", "Doe",
                new BigDecimal("30"), billingAddress, shippingAddress);
  registerTestObject(shippingAddress);
  Product product = new Product(88, "SomeWidget", new BigDecimal("19.99"));
  registerTestObject(shippingAddress);
  Invoice invoice = new Invoice(customer);
  registerTestObject(shippingAddress);
```

Versão Atual do Teste (3)

```
  // Exercise SUT
  invoice.addItemQuantity(product, 5);

  // Verify outcome
  LineItem expected = new LineItem(invoice, product, 5,
                          new BigDecimal("30"), new BigDecimal("69.95"));
  assertContainsExactlyOneLineItem(invoice, expected);
}
```

Exemplo de Teste

E a refatoração continua...

- Refatora o fixture setup criando um método para criar os objetos.

Versão Final do Exemplo de Teste

```
public void testAddItemQuantity_severalQuantity_v14(){
  final int QUANTITY = 5;
  final BigDecimal UNIT_PRICE = new BigDecimal("19.99");
  final BigDecimal CUST_DISCOUNT_PC = new BigDecimal("30");
  // Set up fixture
  Customer customer = createACustomer(CUST_DISCOUNT_PC);
  Product product = createAProduct(UNIT_PRICE);
  Invoice invoice = createInvoice(customer);
  // Exercise SUT
  invoice.addItemQuantity(product, QUANTITY);
  // Verify outcome
  final BigDecimal BASE_PRICE = UNIT_PRICE.multiply(new BigDecimal(QUANTITY));
  final BigDecimal EXTENDED_PRICE = BASE_PRICE.subtract(BASE_PRICE.multiply
        (CUST_DISCOUNT_PC.movePointLeft(2)));
  LineItem expected = createLineItem(QUANTITY, CUST_DISCOUNT_PC,
        EXTENDED_PRICE, product, invoice);
  assertContainsExactlyOneLineItem(invoice, expected);
}
```