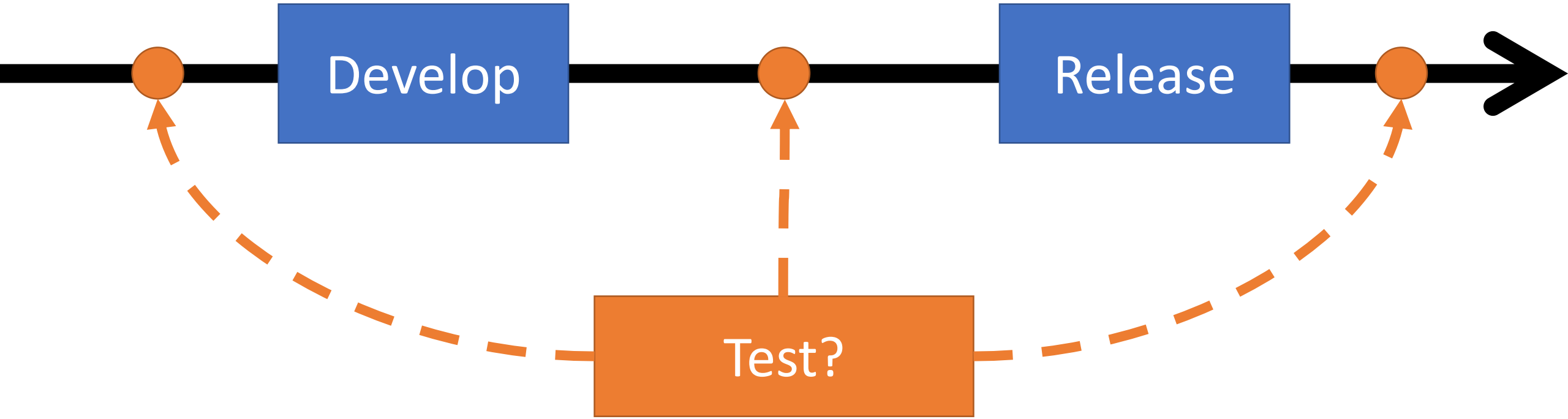
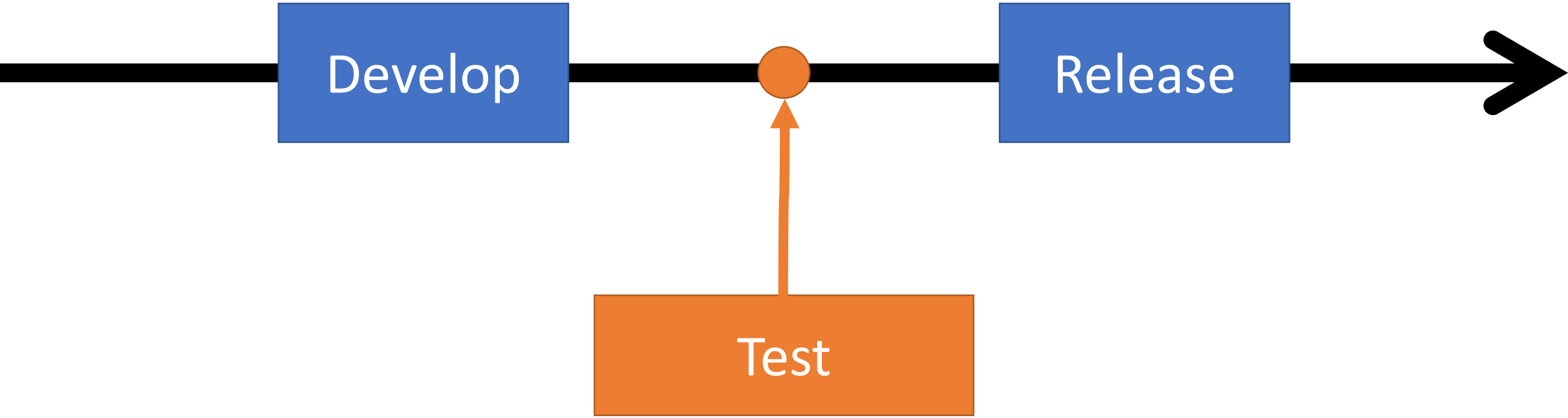
When to test?

Product lifecycle



Testing after development



Testing after development

- Write code
- Write tests
- Fix code
- Yay?

Testing after development

☺ Easiest option

☺ Test when the code is ready

☺ Choose when to stop

Testing after development

☹ Test the implementation only

☹ Too late to fix design

☹ May not test at all

Test-Driven Development (“TDD”)

Test-Driven Development

- Write **tests**
- Write code
- Fix code
- Yay?

Test-Driven Development

☺ Think before coding

☺ Less debugging needed

☺ Instant feedback when coding

Test-Driven Development

☹ Higher time investment

☹ Waste time if design changes a lot

TDD Example

“Users can withdraw money

from their bank account,

but only as much as they have.”

TDD Example

“Can a bank account have

a balance below 0?”

“No.”

TDD Example

```
class Account {
}
```

TDD Example

```
@Test
void canWithdrawNothing() {
   Account account = new Account(100);
   assertThat(account.withdraw(0), is(0));
}
```

TDD Example

```
class Account {
   public Account(int balance) { }

   int withdraw(int amount) { return -1; }
}
```

TDD Example

```
@Test
void cannotInitializeWithNegativeBalance() {
    assertThrows(IllegalArgumentException.class,
                 () -> new Account(-1));
}
```

TDD Example

```
@Test
void canWithdrawLessThanBalance() {
   Account account = new Account(100);
   assertThat(account.withdraw(10), is(10));
   assertThat(account.balance(), is(90));
}
```

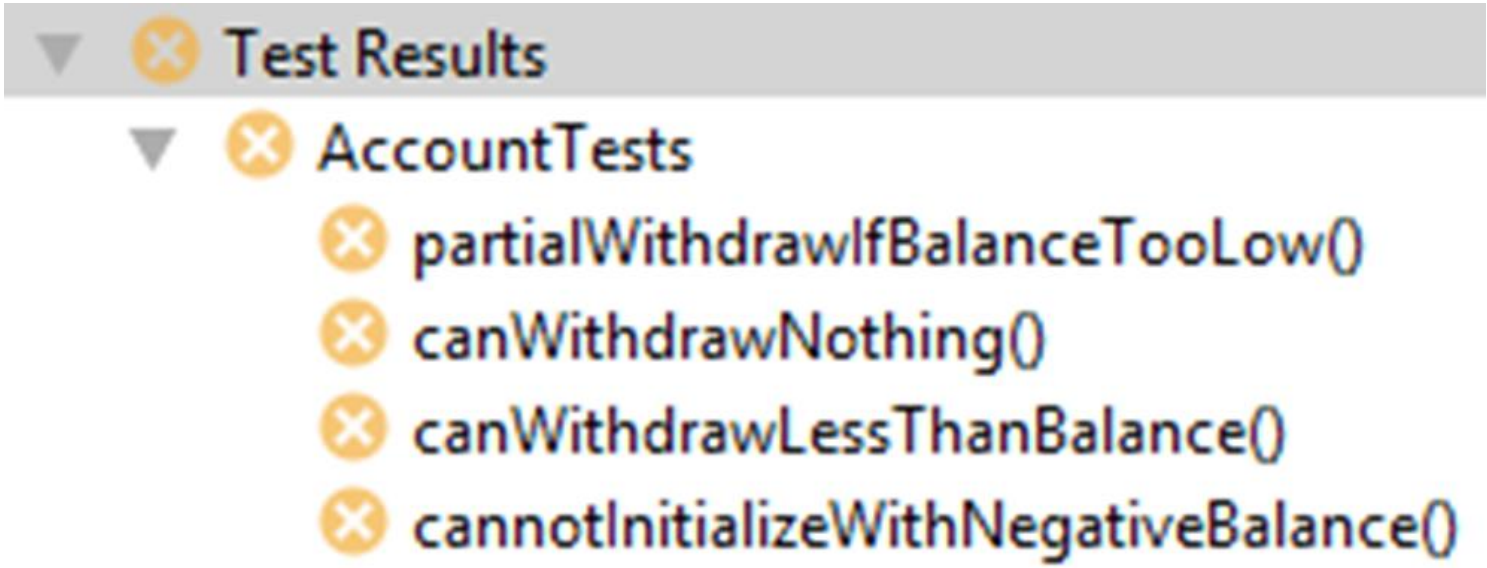
TDD Example

```
class Account {
   int balance() { return -1; }

   public Account(int balance) { }

   int withdraw(int amount) { return -1; }
}
```

TDD Example

```
@Test
void partialWithdrawIfBalanceTooLow() {
   Account account = new Account(10);
   assertThat(account.withdraw(20), is(10));
   assertThat(account.balance(), is(0));
}
```

TDD Example


Test Results
AccountTests
partialWithdrawIfBalanceTooLow()
canWithdrawNothing()
canWithdrawLessThanBalance()
cannotInitializeWithNegativeBalance()

TDD Example

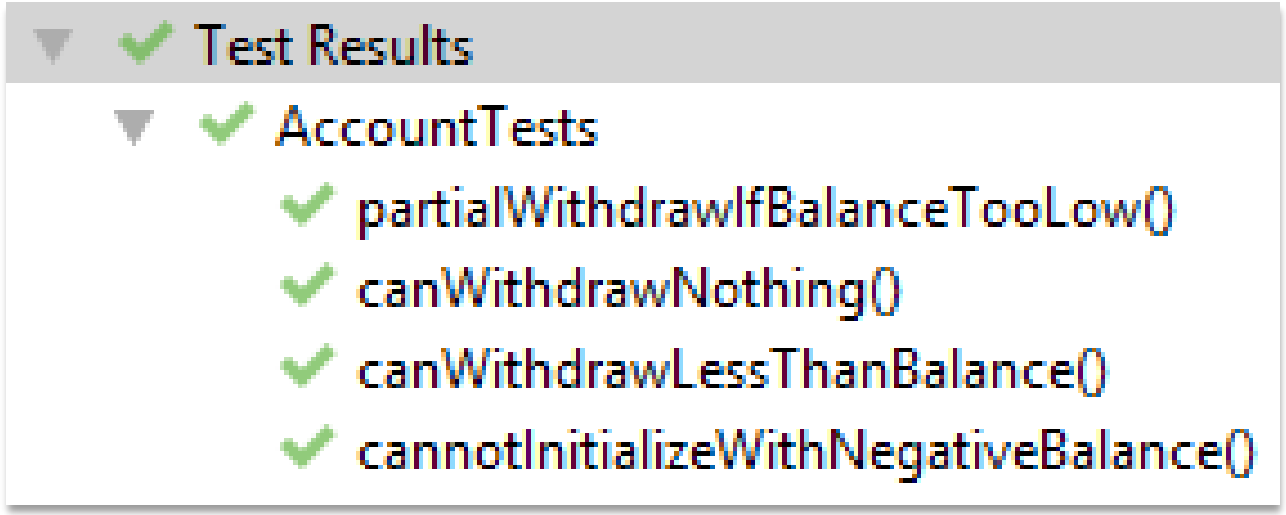
```
private int balance;

public int balance() { return balance; }
```

TDD Example

```
public Account(int balance) {
   if (balance < 0) {
      throw new IllegalArgumentException();
   }

   this.balance = balance;
}
```

TDD Example

```
public int withdraw(int amount) {
    int result = Math.min(balance, amount);
    balance -= result;
    return result;
}
```



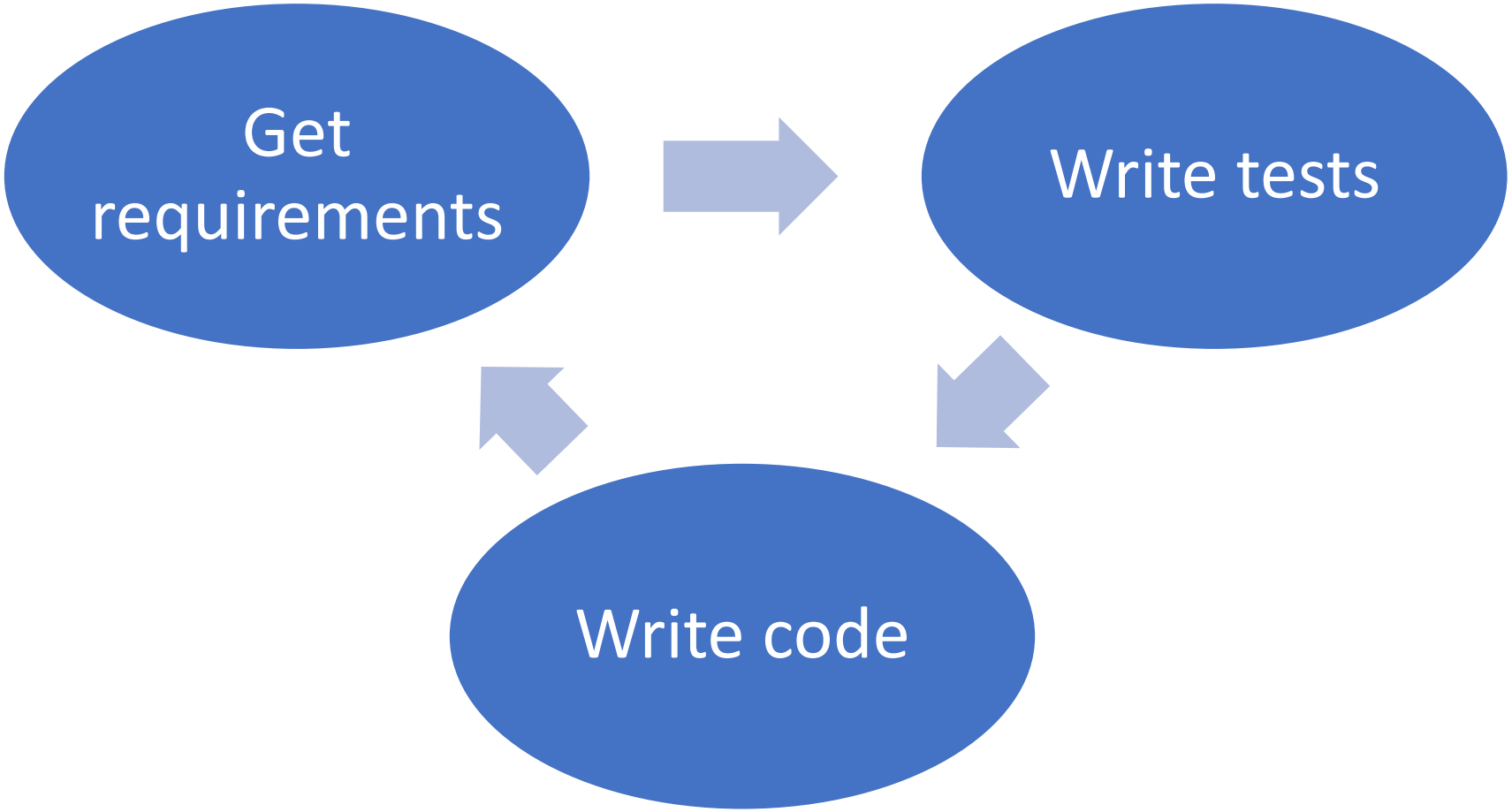
TDD Example

“We forgot!

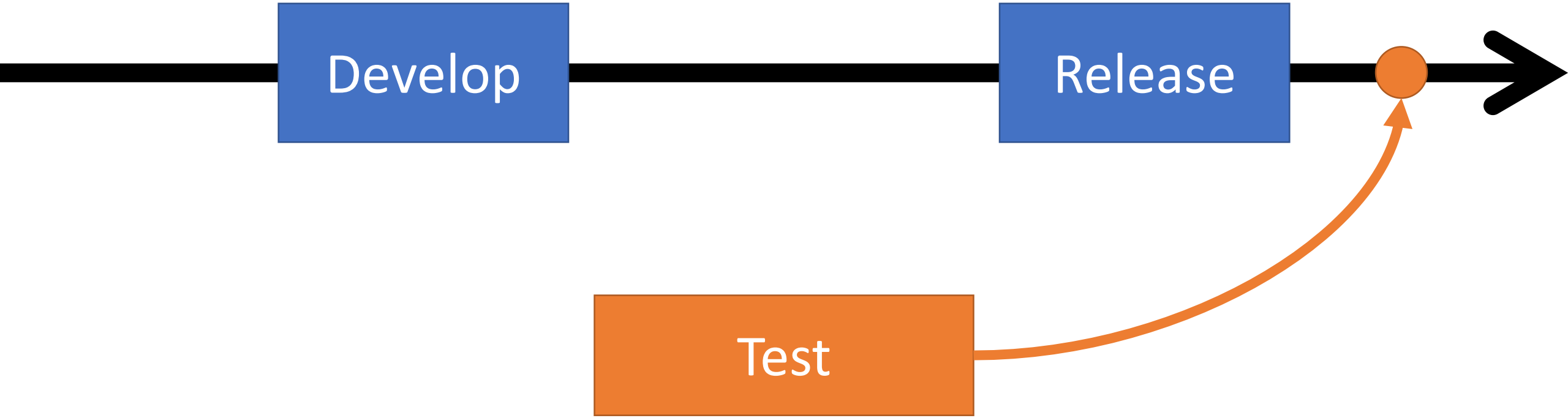
Accounts can be blocked.

Withdrawing from a blocked account has no effect.”

TDD Recap



Regression testing



Regression testing

- You will have bugs
- How to deal with them?

Regression testing

- Reproduce the bug
- Fix the bug
- Confirm the fix works
- How?

Regression testing

- Reproduce the bug with a test
- Fix the bug
- Confirm the fix works by running the test

Regression testing at Microsoft

Quick info cannot show type for linq group variable declaration #38794

Closed huoyaoyuan opened this issue 16 days ago · 0 comments · Fixed by #39089

huoyaoyuan commented 16 days ago Contributor

Version Used:

VS 16.3 Preview 4

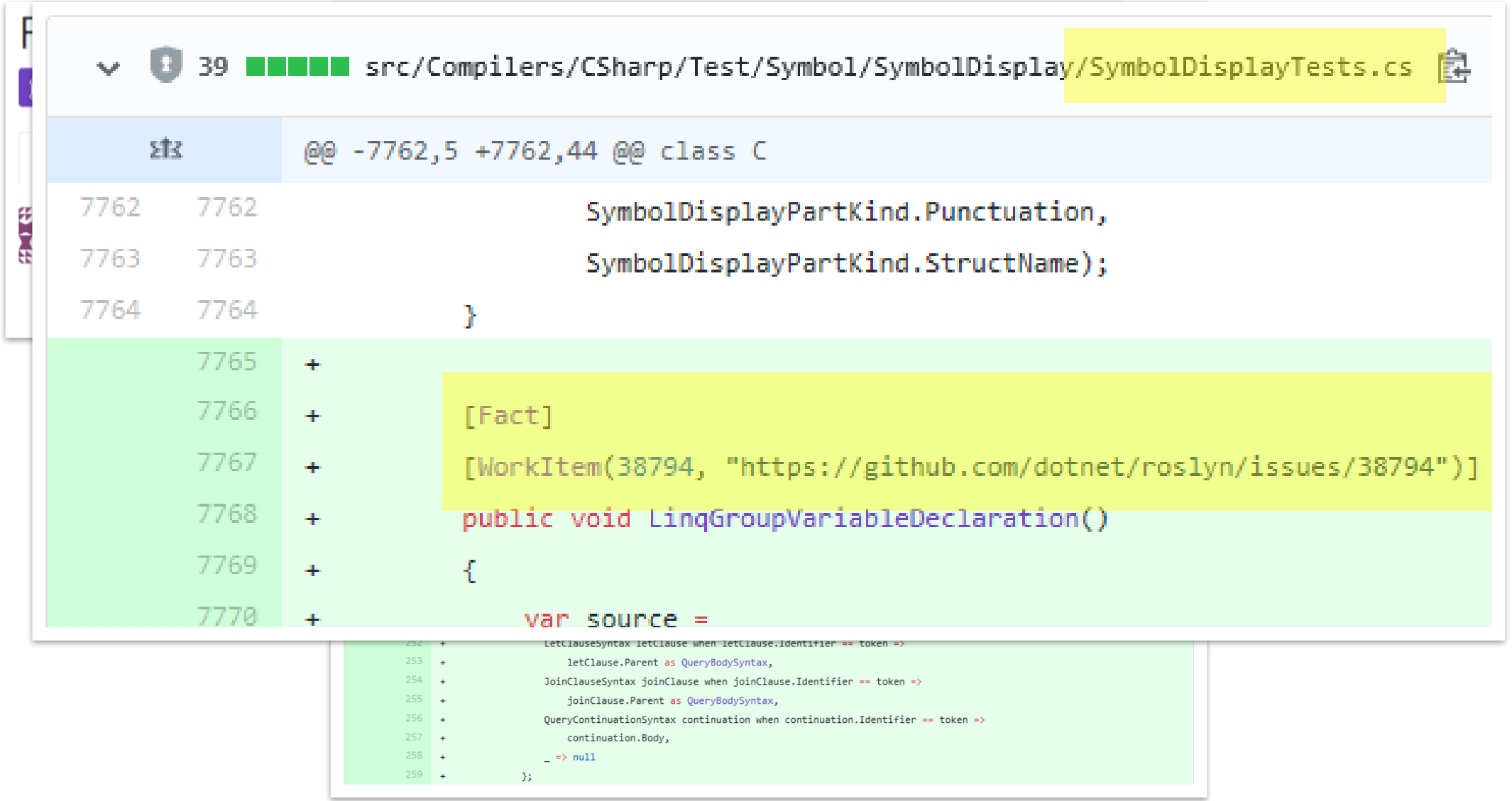
```
void M(string[] a)
{
    var v = from x in a
            group x by x.Length into g
            select (g.Key, g);
}
```

(范围变量) ? g

```
void M(string[] a)
{
    var v = from x in a
            group x by x.Length into g
            select (g.Key, g);
}
```

(范围变量) IGrouping<int, string> g

Regression testing at Microsoft



Regression testing at Microsoft