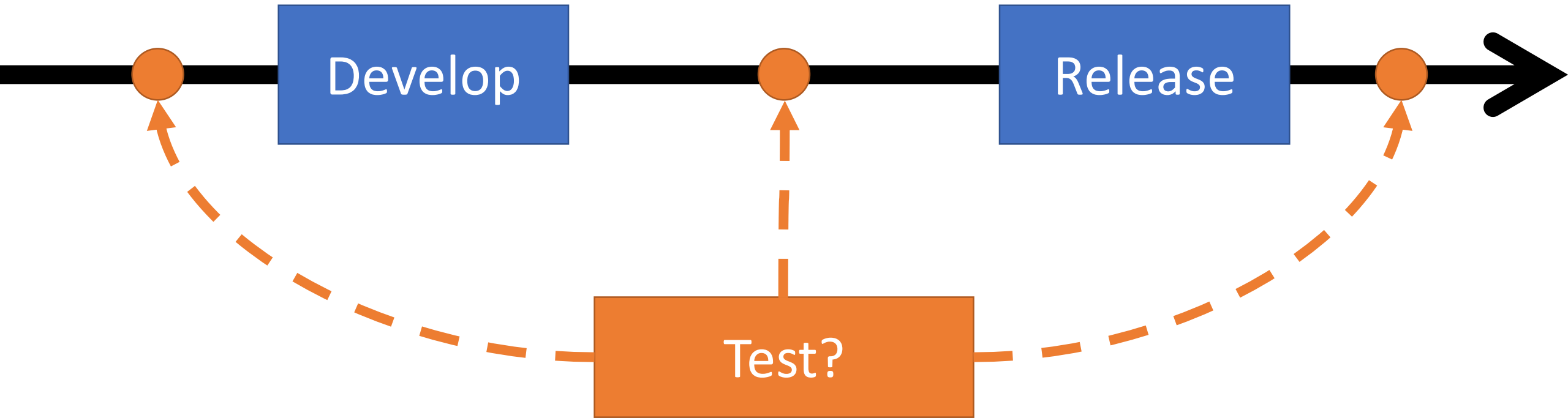
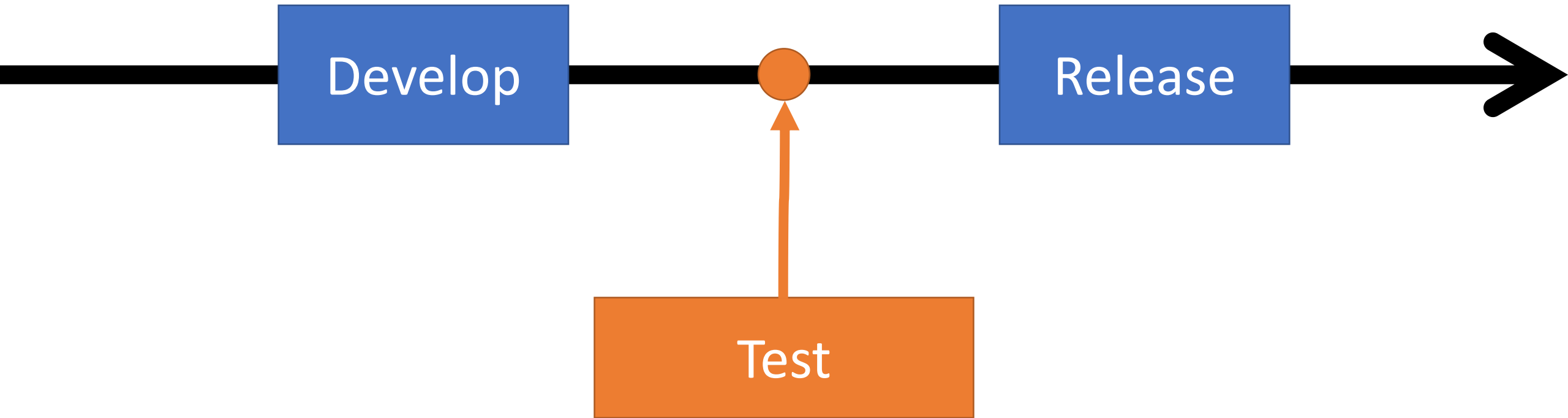


When to test?

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

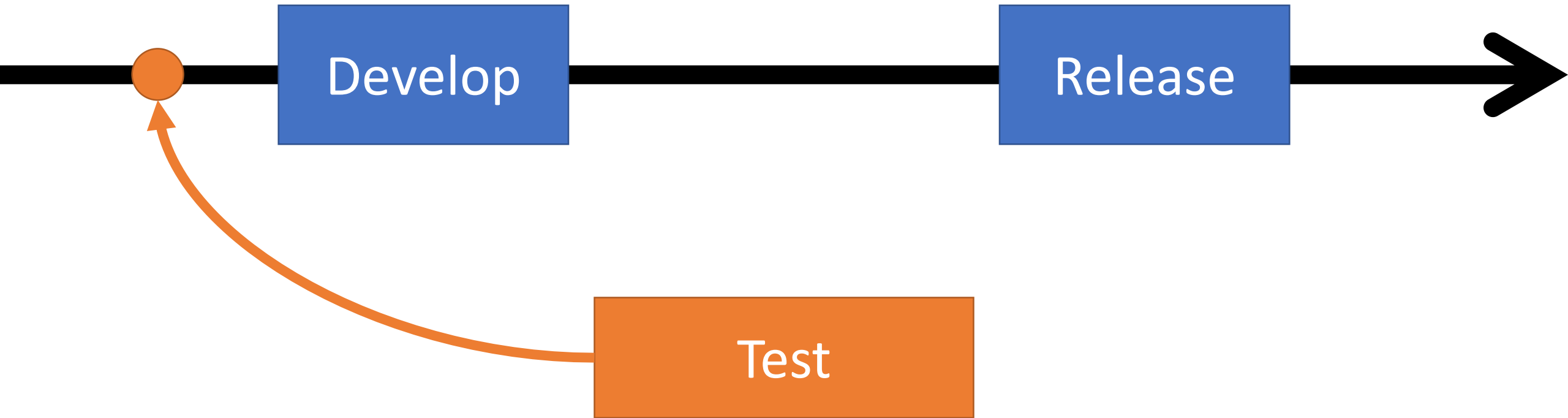
Testing after development

☹️ 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

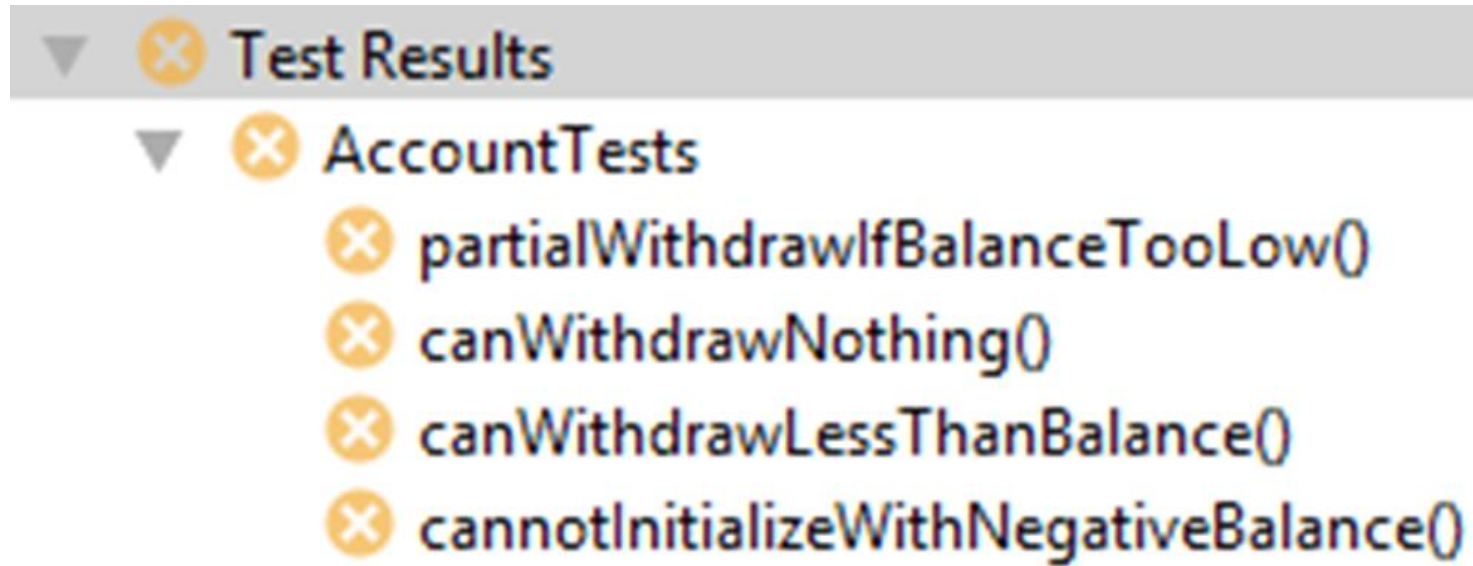
```
class Account {  
    int balance() { return -1; }  
  
    public Account(int balance) { }  
  
    int withdraw(int amount) { return -1; }  
}
```

TDD Example

@Test

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

TDD Example



TDD Example

```
private int balance;
```

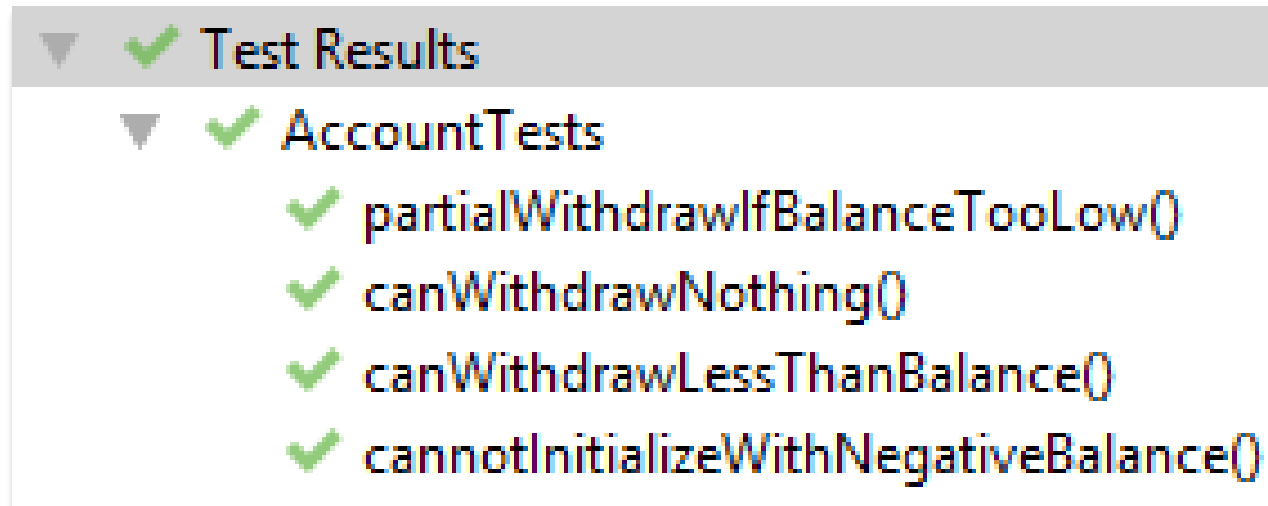
```
public int balance() { return balance; }
```

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;  
}
```



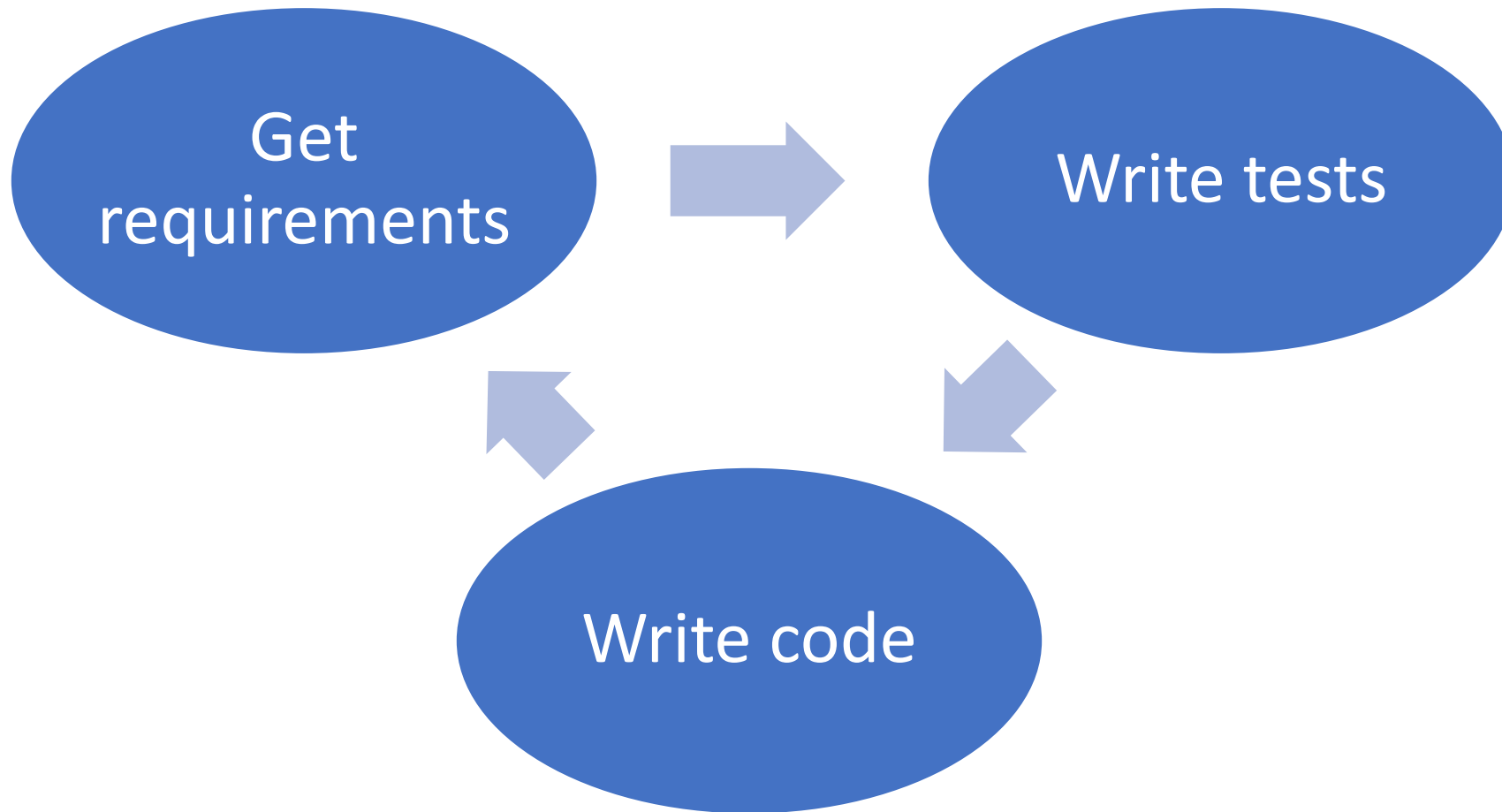
TDD Example

“We forgot!

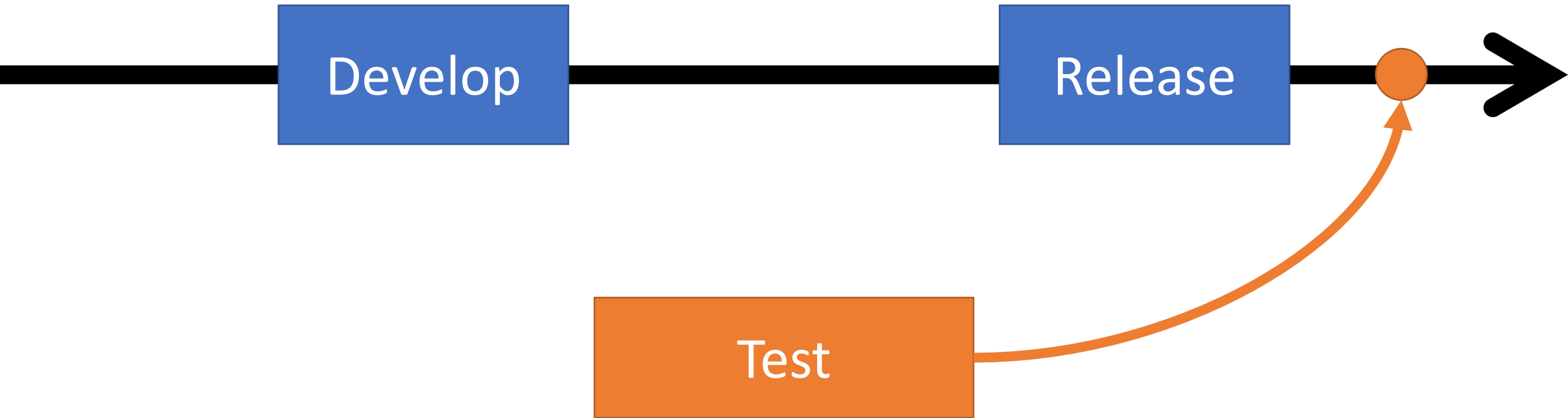
Accounts can be blocked.

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

▼ ⓘ 39 ■■■■■ src/Compilers/CSharp/Test/Symbol/SymbolDisplay/SymbolDisplayTests.cs

Line	Diff	Code
7762	7762	SymbolDisplayPartKind.Punctuation,
7763	7763	SymbolDisplayPartKind.StructName);
7764	7764	}
7765	+	
7766	+	[Fact]
7767	+	[WorkItem(38794, "https://github.com/dotnet/roslyn/issues/38794")]
7768	+	public void LinqGroupVariableDeclaration()
7769	+	{
7770	+	var source =

252	+	LetClauseSyntax letClause when letClause.Identifier == token =>
253	+	letClause.Parent as QueryBodySyntax,
254	+	JoinClauseSyntax joinClause when joinClause.Identifier == token =>
255	+	joinClause.Parent as QueryBodySyntax,
256	+	QueryContinuationSyntax continuation when continuation.Identifier == token =>
257	+	continuation.Body,
258	+	- => null
259	+	};

Regression testing at Microsoft

  **RikkiGibson** merged commit `6dab121` into `dotnet:master` 21 hours ago

18 checks passed

✓ WIP Ready for review	Details
✓ license/ccla All CLA requirements met.	Details
✓ roslyn-CI Build #20191006.3 succeeded	Details
✓ roslyn-CI (Linux_Test coreclr) Linux_Test coreclr succeeded	Details
✓ roslyn-CI (SourceBuild_Test) SourceBuild_Test succeeded	Details
✓ roslyn-CI (Windows_CoreClr_Unit_Tests debug) Windows_CoreClr_Unit_Tests d...	Details
✓ roslyn-CI (Windows_CoreClr_Unit_Tests release) Windows_CoreClr_Unit_Tests r...	Details
✓ roslyn-CI (Windows_Correctness_Test) Windows_Correctness_Test succeeded	Details
✓ roslyn-CI (Windows Desktop Spanish Unit Tests) Windows Desktop Spanish...	Details