Unit Testing Makes Me Faster

Convincing Your Boss, Your Coworkers, and Yourself

Jeremy Clark
www.jeremybytes.com

Bosses Hate Tests

Production Code

```
 1  using ...
10
11  namespace Module.Catalog
12  {
13      public class CatalogViewModel : INotifyPropertyChanged
14      {
15          Fields
30
31          Properties
126
127         Constructors
135
136         Methods
251
252         INotifyPropertyChanged Members
263     }
264 }
265
```

Test Code

```
 1  using ...
11
12  namespace Module.Catalog.Test
13  {
14      [TestClass]
15      public class CatalogViewModelTest
16      {
17          Test Initialization
116
117         Model Initialization
148
149         Catalog Population
182
183         Service Exception
240
241         Catalog Caching
287
288         Filters
355
356         Filter Reset
418
419         Catalog Item Selection
534     }
535 }
536
```

Is Typing Really Our Limitation?

Different Kinds of Tests

- Unit Testing
- Integration Testing
- Performance Testing
- Exploratory Test
- Penetration Testing
- User Acceptance Testing (UAT)

What are Unit Tests?

A unit test is an automated piece of code that invokes a unit of work in the system and then checks a single assumption about the behavior of that unit of work.

The Art of Unit Testing by Roy Osherove

Non-Threatening Text Here

Threatening
Text Here

What are Unit Tests?

A unit test is an automated piece of code that invokes a unit of work in the system and then checks a single assumption about the behavior of that unit of work.

The Art of Unit Testing by Roy Osherove

automated piece of code

a unit of work

checks a single assumption

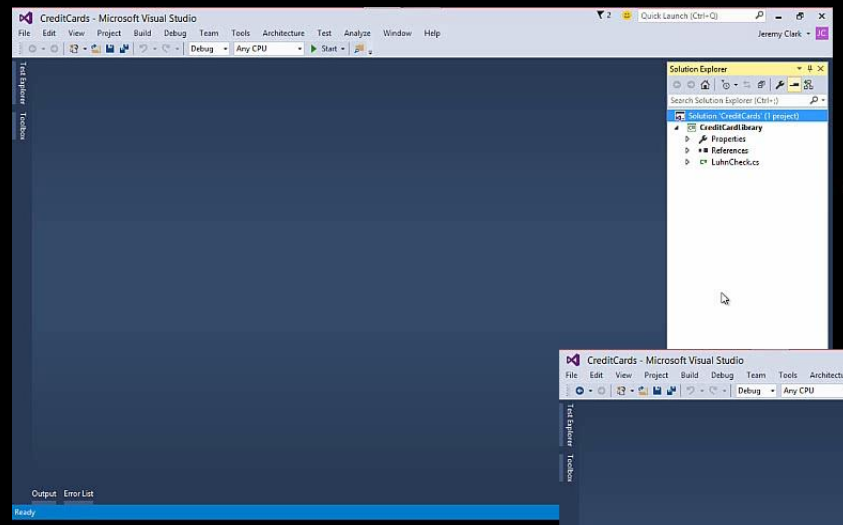
What Makes Me Faster?

- Confirming Functionality
- Checking Regression
- Pinpointing Bugs
- Documenting Functionality

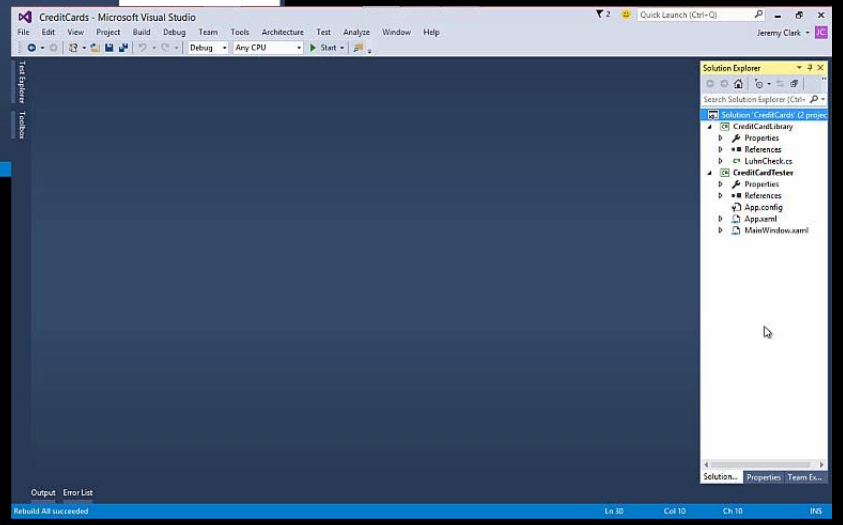
Confirming Functionality

Unit Tests are ***proof*** that my code does what I ***think*** it does

Build Time Comparison



Test Application

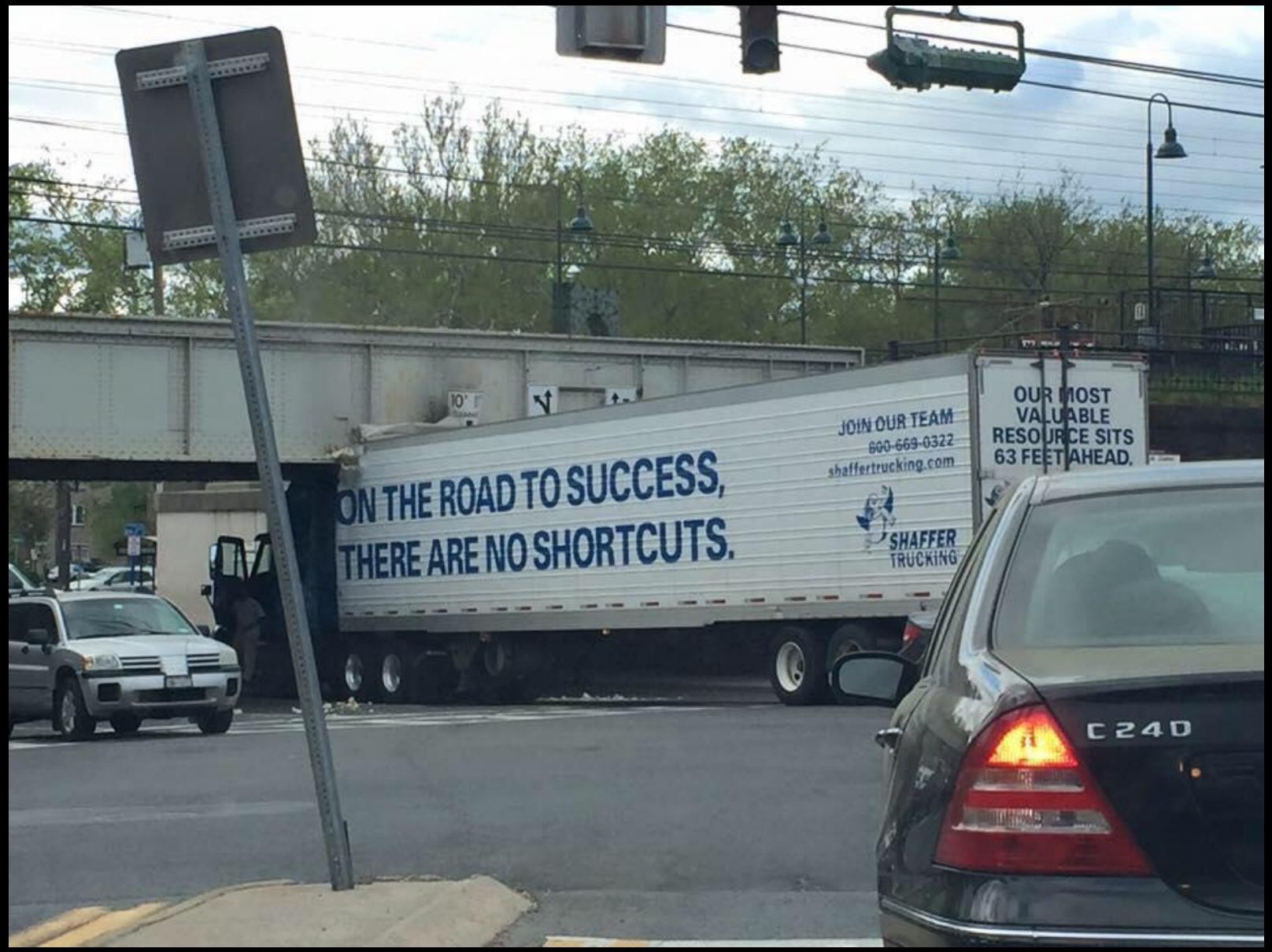


Unit Testing

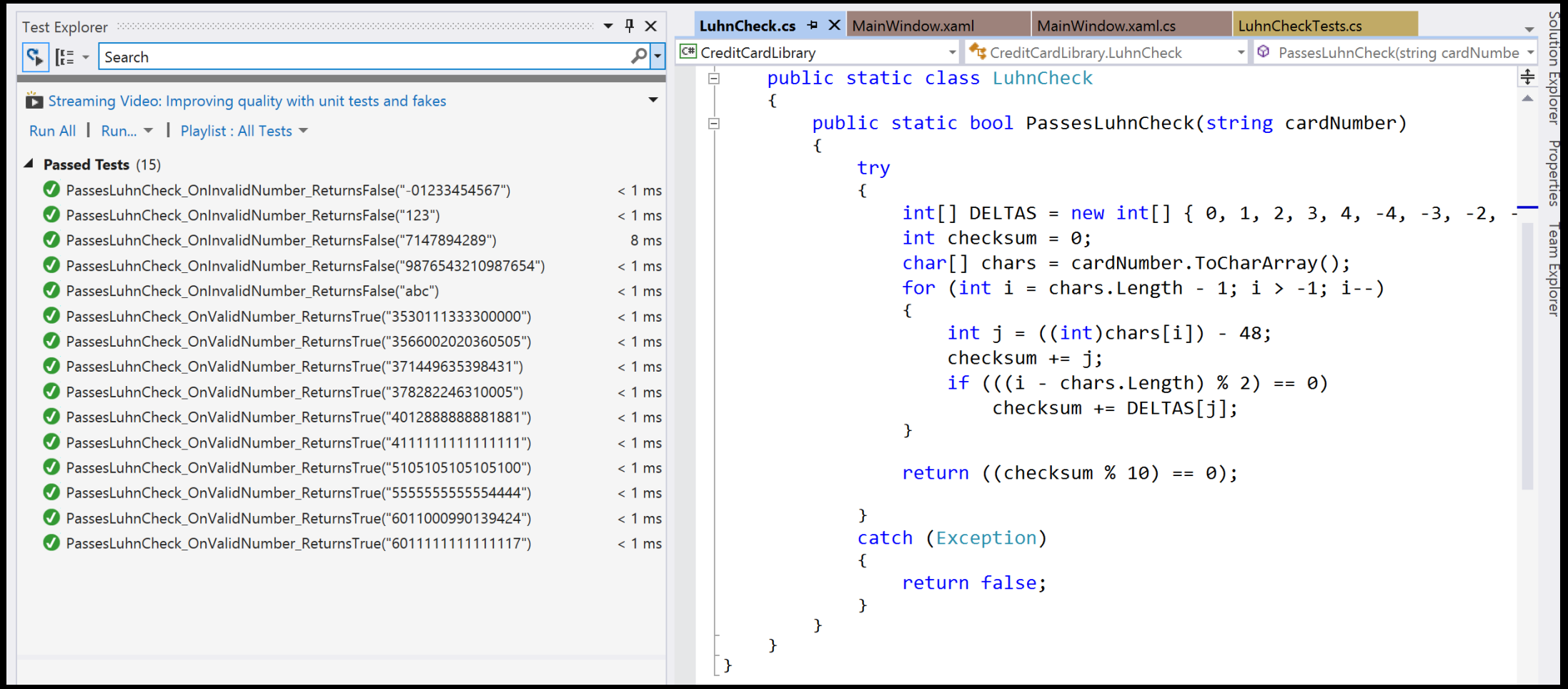
Disclaimer

We get these advantages when we are ***comfortable*** writing ***good*** tests.

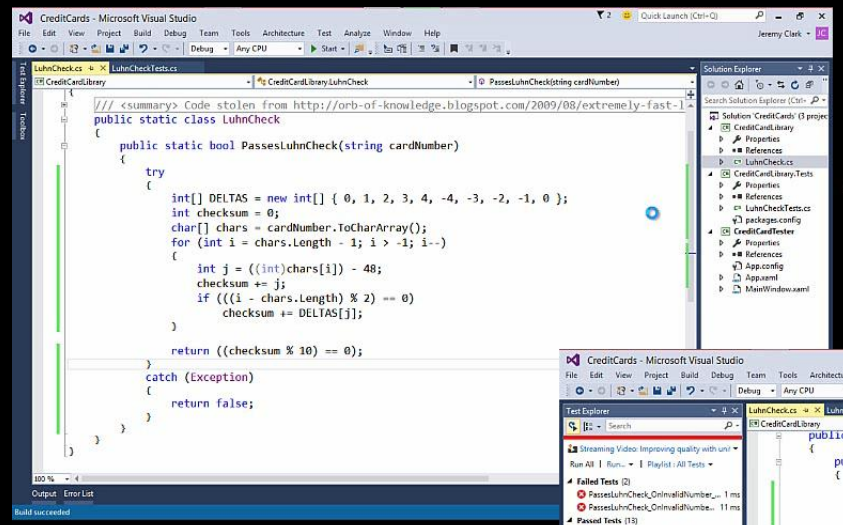
Realistic Expectations



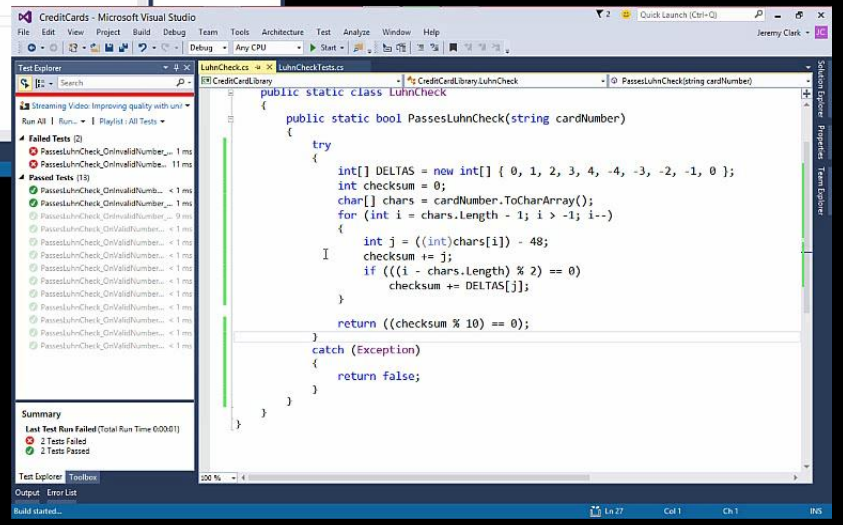
Checking Regression



Regression Comparison

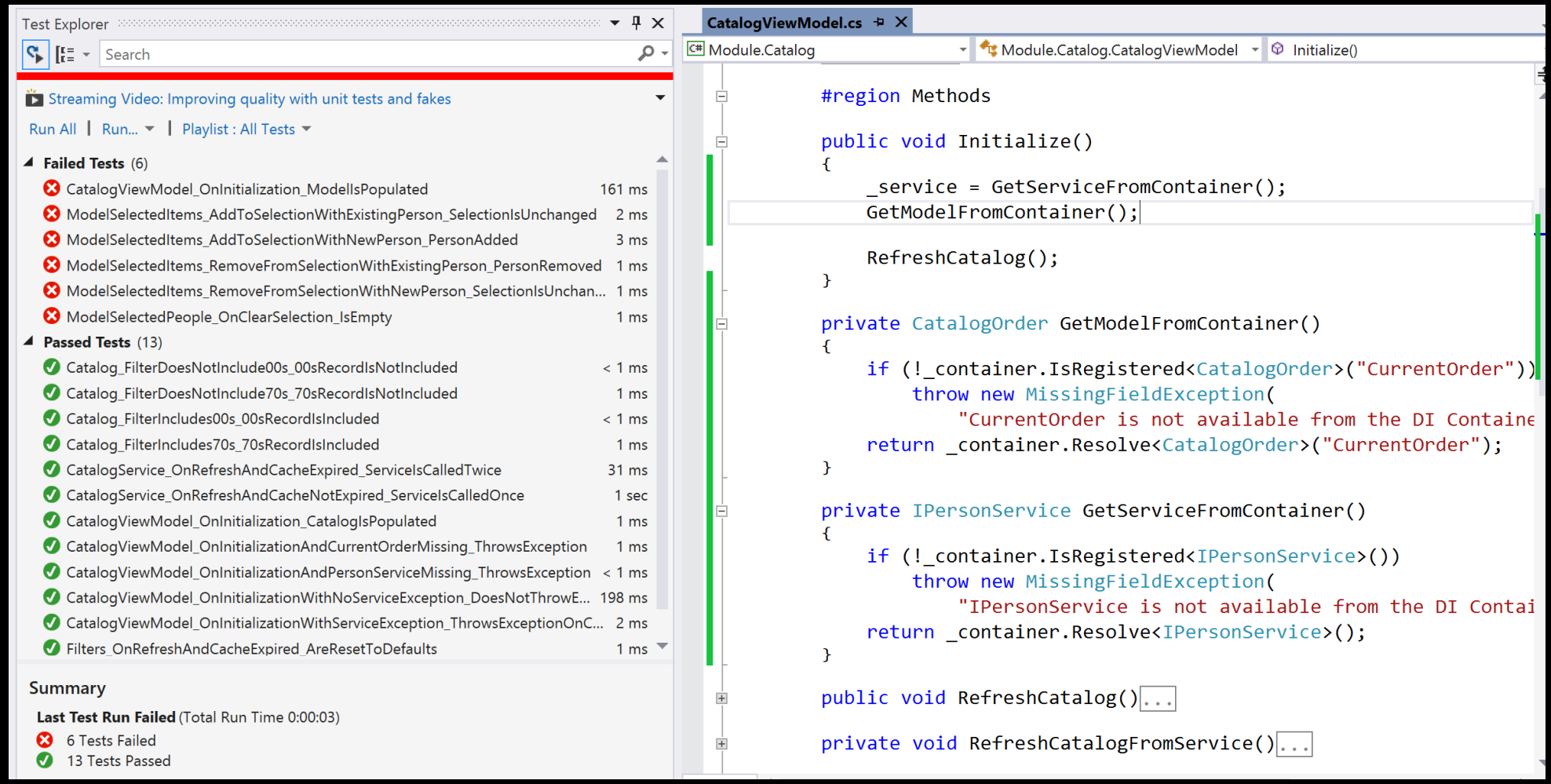


Test Application

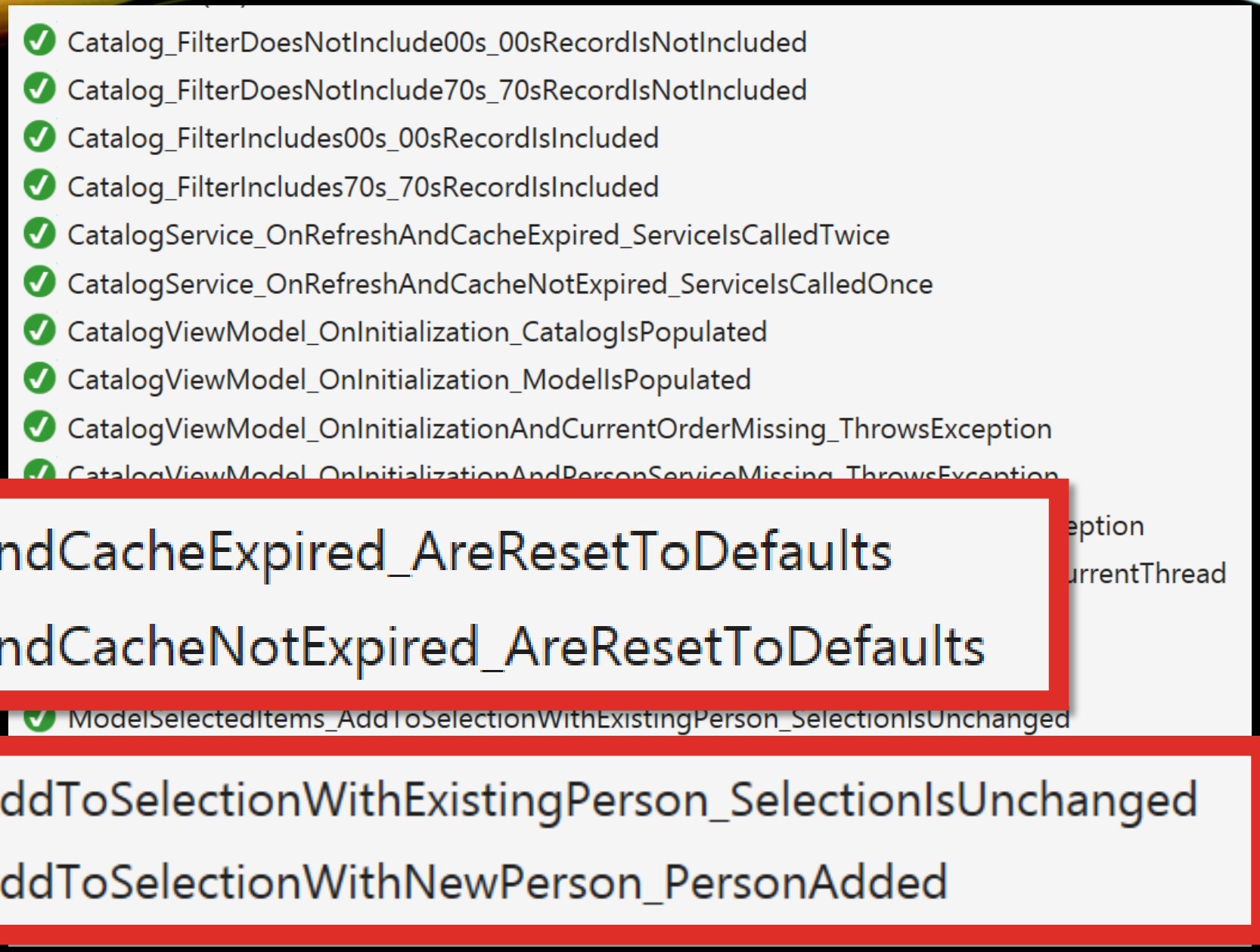


Unit Testing

Pinpointing Bugs



Documenting Functionality



- Filters_OnRefreshAndCacheExpired_AreResetToDefaults
- Filters_OnRefreshAndCacheNotExpired_AreResetToDefaults

- ModelSelectedItems_AddToSelectionWithExistingPerson_SelectionIsUnchanged
- ModelSelectedItems_AddToSelectionWithNewPerson_PersonAdded

Disclaimer

We get these advantages when we are ***comfortable*** writing ***good*** tests.

Good Unit Tests

- Maintainable
- Dependable
- Runnable

Qualities of a Good Test

Maintainable

- Not Tricky
- Easy to Read
- Easy to Write
- Well-Named

Dependable

- Consistent Results
- Isolated
- Continued Relevance
- Tests the Right Things

Runnable

- FAST

Michael C. Feathers on Speed

"A unit test that takes 1/10th of a second to run is a slow unit test."

"Unit tests run fast. If they don't run fast, they aren't unit tests."

Working Effectively with Legacy Code by Michael C. Feathers

Qualities of a Good Test

Maintainable

- Not Tricky
- Easy to Read
- Easy to Write
- Well-Named

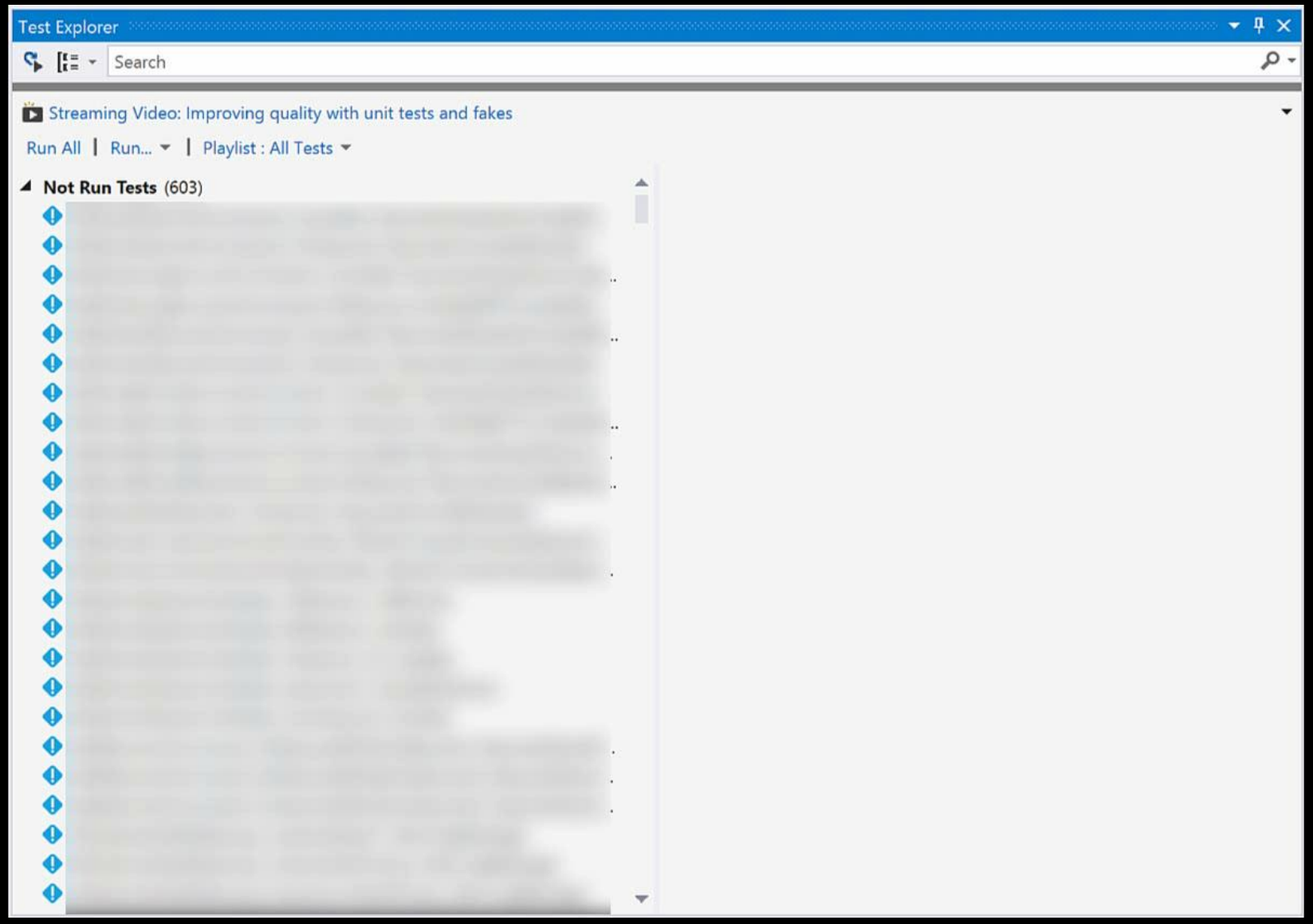
Dependable

- Consistent Results
- Isolated
- Continued Relevance
- Tests the Right Things

Runnable

- FAST
- Single Click
- Repeatable
- Failure Points to the Problem

Isolated and Fast



Code Coverage

100% Code Coverage is not a guarantee

Conversations about Code Coverage

"What parts of your application are okay ***not*** to test?"

The Stahl Standard

"What parts of your application do your users ***not*** care about?"

-Barry Stahl

Twitter: @bsstahl http://www.cognitiveinheritance.com/

Know the Goals

- Don't do the right thing for the wrong reason.
- Unit testing will not fix bad development practices.

http://www.jenders.com/2012/01/08/thief-almost-caught-on-camera-stealing-thin-lg-television/

Martin Fowler on Fear

"Don't let the fear that testing can't catch all bugs stop you from writing the tests that will catch most bugs."

Refactoring by Martin Fowler et al.

References

- *The Art of Unit Testing with Examples in C#* – Roy Osherove
- *Refactoring* – Martin Fowler et al.
- *Working Effectively with Legacy Code* – Michael C. Feathers

- *Test-Driven Development by Example* – Kent Beck
- *Refactoring to Patterns* – Joshua Kerievsky
- *Agile Principles, Patterns, and Practices in C#* – Robert C. Martin & Micah Martin
- *Code Complete* – Steve McConnell
- *Beautiful Testing* – Edited by Tim Riley & Adam Goucher

What Makes Me Faster?

- Confirming Functionality
- Checking Regression
- Pinpointing Bugs
- Documenting Functionality

Thank You!

Jeremy Clark

- http://www.jeremybytes.com
- jeremy@jeremybytes.com
- @jeremybytes