Taming Asynchronous .NET Code with Rx

INTRODUCING REACTIVE PROGRAMMING

Edin Kapić

@ekapic www.edinkapic.com

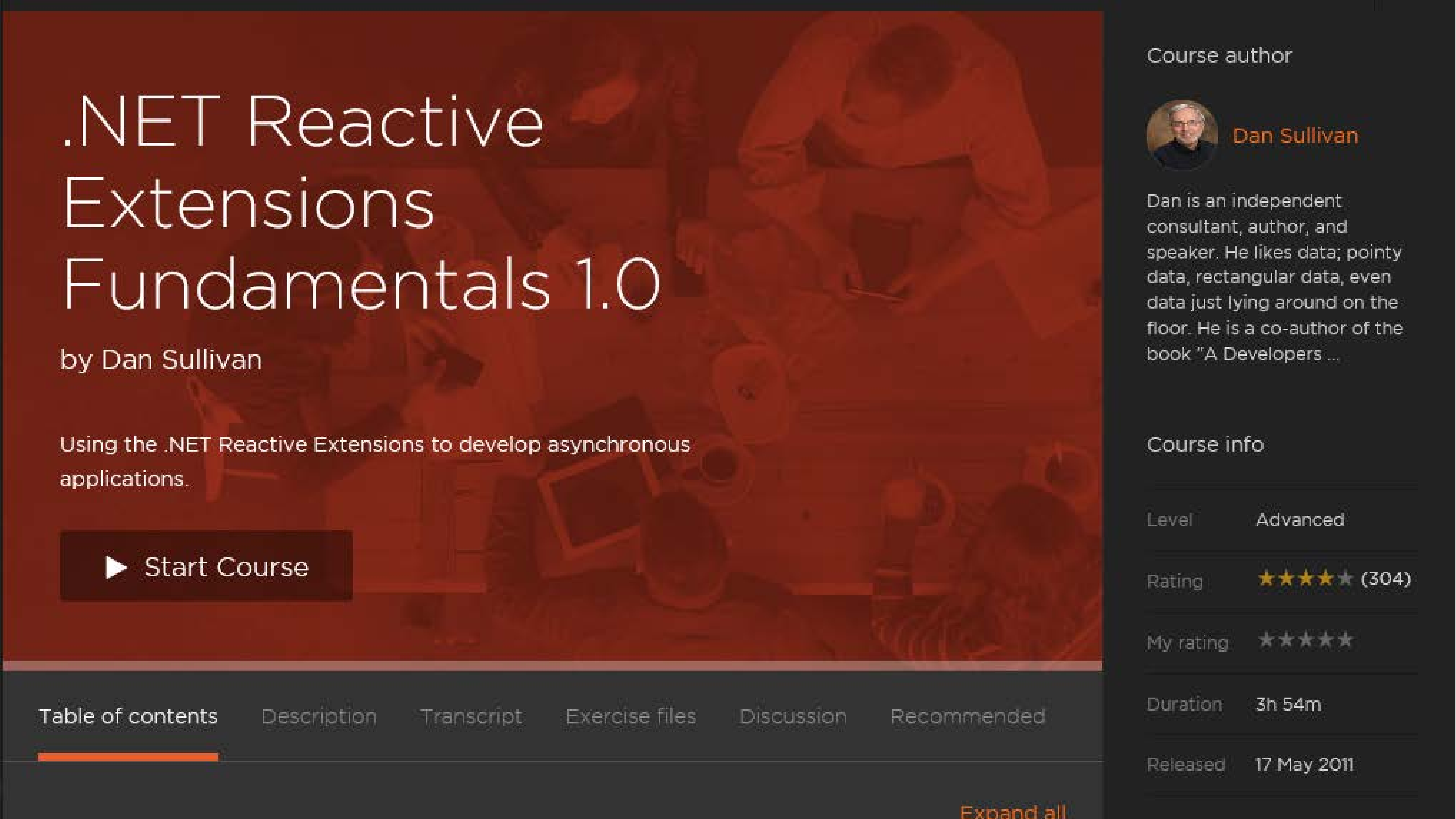
Asynchronous and event-driven programming is difficult

Reactive Extensions (Rx)

Demo

“Show a button if the mouse hovers for more than half a second in the designated active area”

- With event handlers
- With Rx sequences



.NET Reactive Extensions Fundamentals 1.0

by Dan Sullivan

Using the .NET Reactive Extensions to develop asynchronous applications.

Course author

Dan Sullivan

Dan is an independent consultant, author, and speaker. He likes data; pointy data, rectangular data, even data just lying around on the floor. He is a co-author of the book "A Developers ...

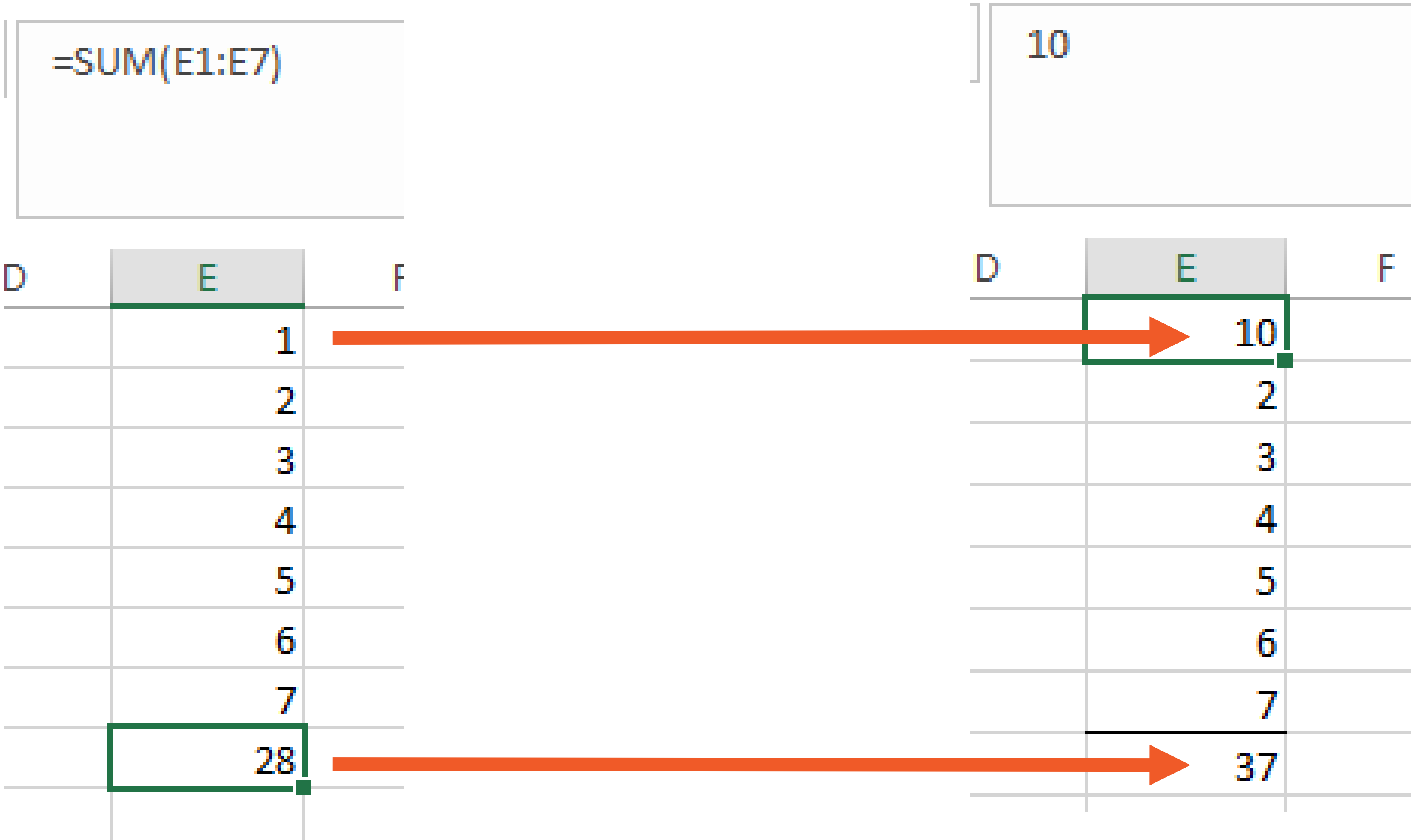
Course info

Level	Advanced
Rating	★★★★★ (304)
My rating	★★★★★
Duration	3h 54m
Released	17 May 2011

“Reactive Programming is a programming paradigm oriented around **data flows** and the **propagation of change**”

Wikipedia

Reactive Excel Spreadsheets



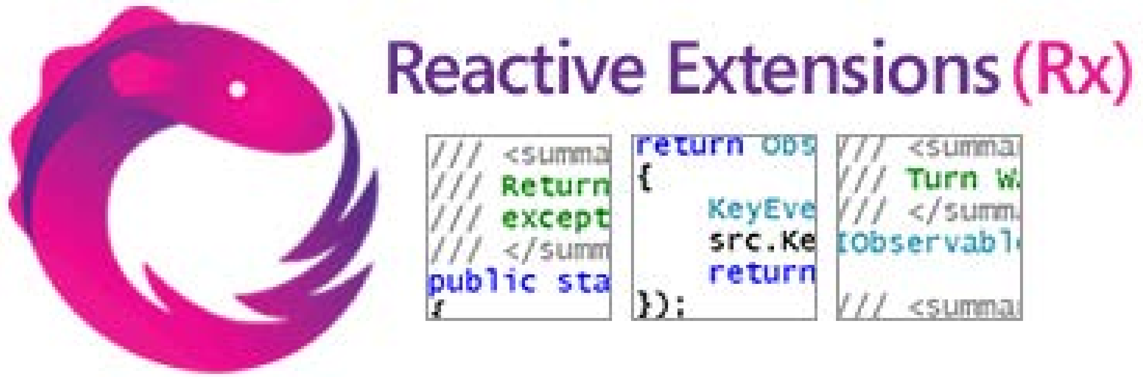
=SUM(E1:E7)

D	E	F
	1	
	2	
	3	
	4	
	5	
	6	
	7	
	28	

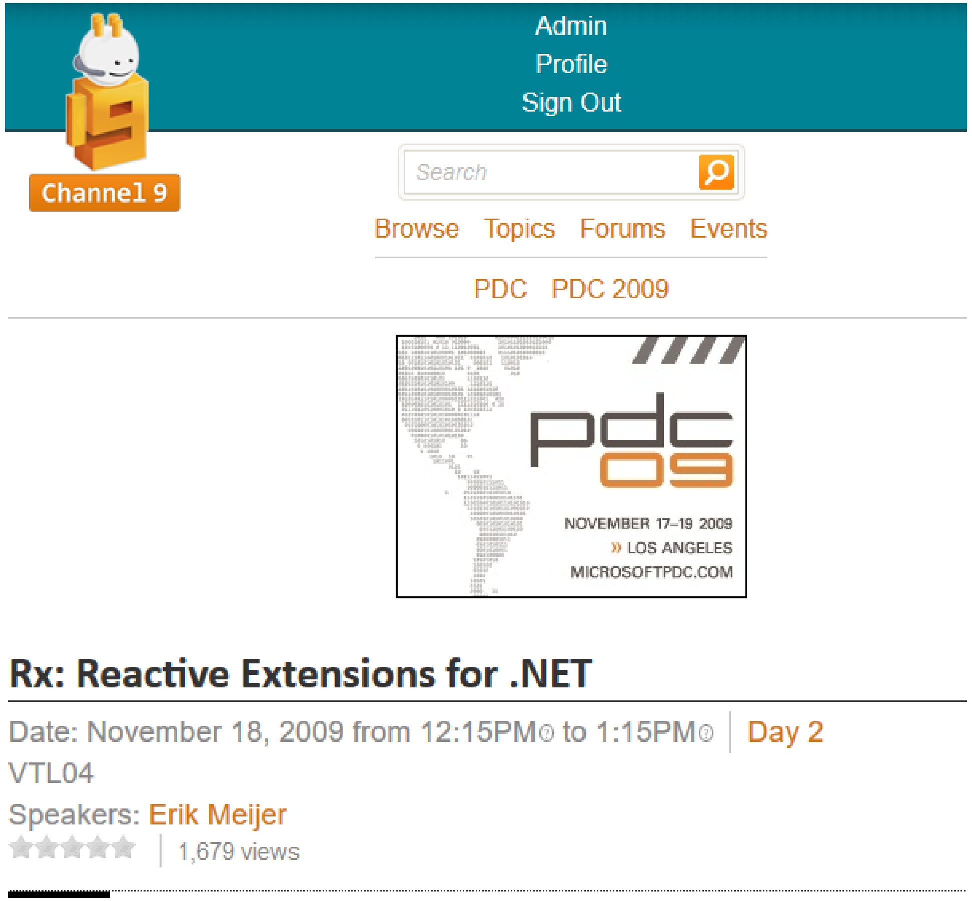
10

D	E	F
	10	
	2	
	3	
	4	
	5	
	6	
	7	
	37	

Reactive Extensions (Rx) Library



https://github.com/Reactive-Extensions/Rx.NET



Rx: Reactive Extensions for .NET

Date: November 18, 2009 from 12:15PM to 1:15PM | Day 2

VTL04

Speakers: Erik Meijer

☆☆☆☆☆ | 1,679 views

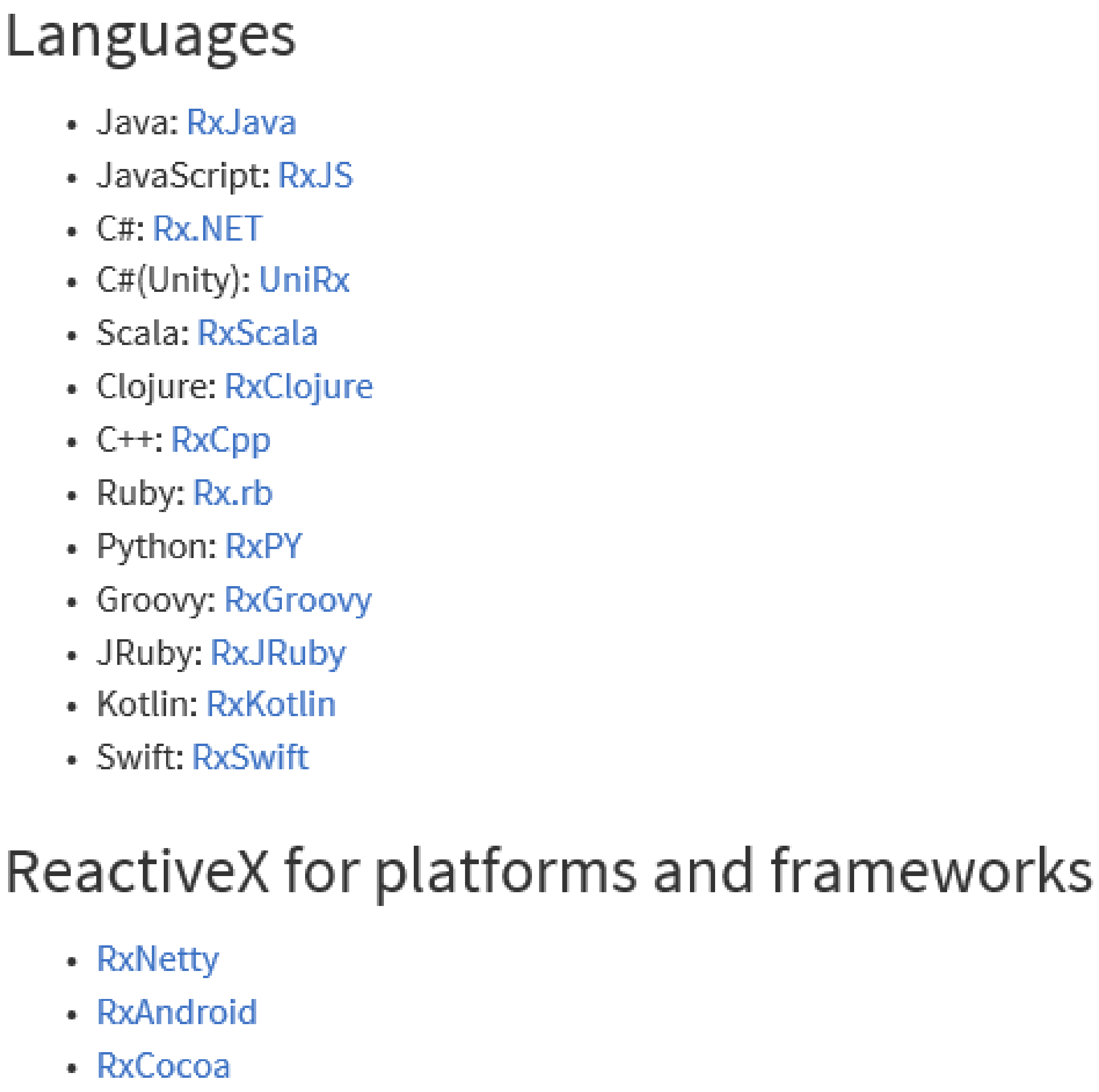
Rx Library Pillars

Flows, Sequences and Events

LINQ and Extensions

Schedulers and Concurrency

Rx Availability



Languages

- Java: RxJava
- JavaScript: RxJS
- C#: Rx.NET
- C#(Unity): UniRx
- Scala: RxScala
- Clojure: RxClojure
- C++: RxCpp
- Ruby: Rx.rb
- Python: RxPY
- Groovy: RxGroovy
- JRuby: RxJRuby
- Kotlin: RxKotlin
- Swift: RxSwift

ReactiveX for platforms and frameworks

- RxNetty
- RxAndroid
- RxCocoa

An API for asynchronous programming with observable streams

http://reactivex.io

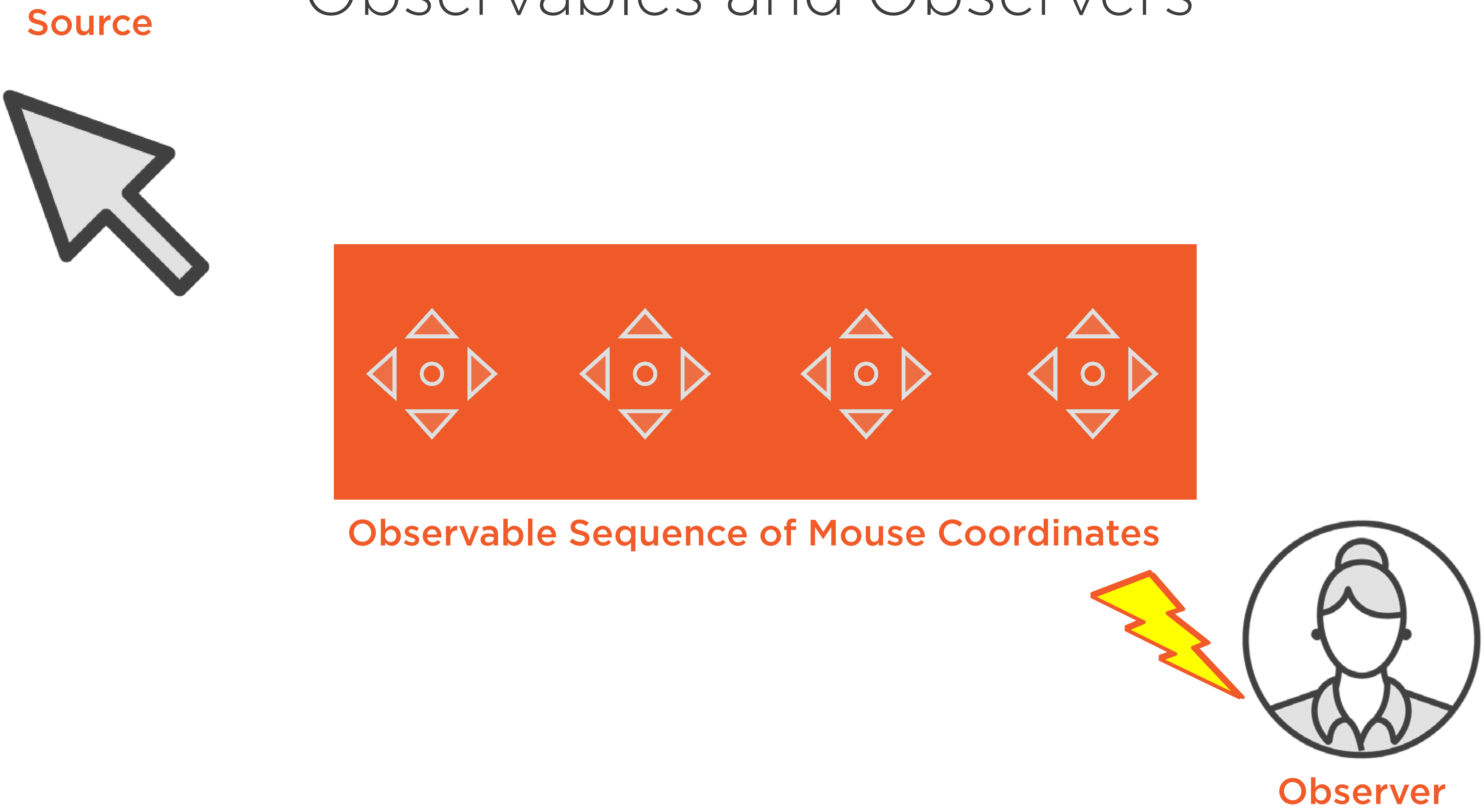
Events in .NET

Implementation of Observer pattern with delegates

Not a first-class citizen in .NET

- Clumsy syntax (+=)
- Difficult to pass around
- Awkward chaining and error handling
- No history of events

Observables and Observers



Source

Observable Sequence of Mouse Coordinates

Observer

Core Interfaces

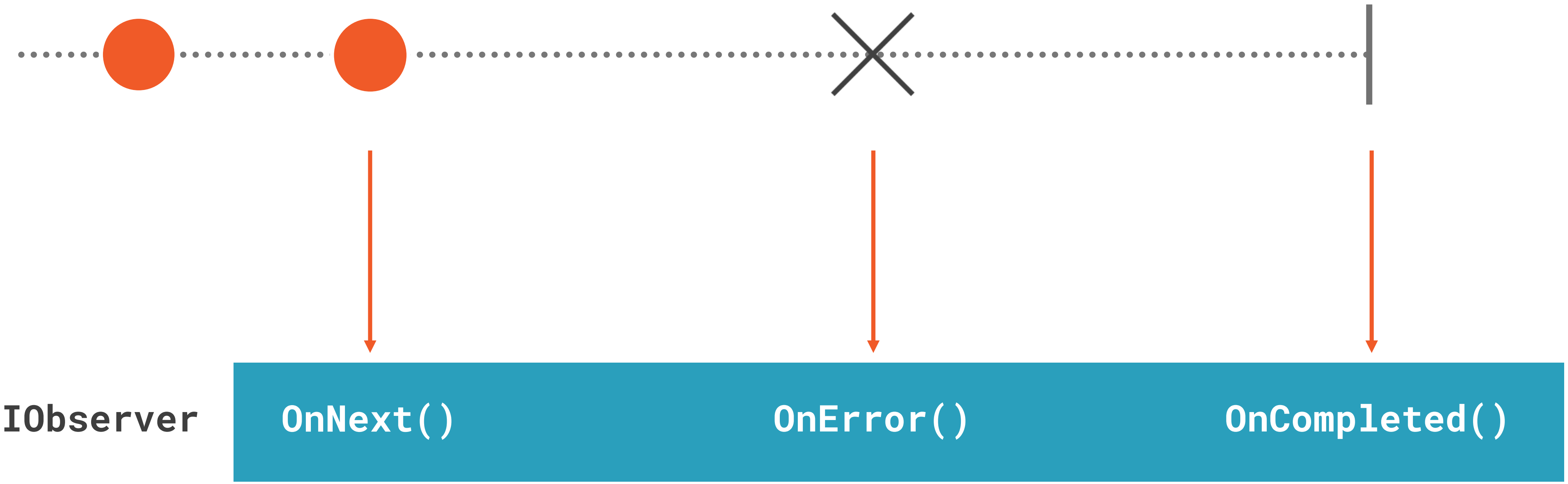
IObservable

```
Subscribe(IObserver<T> observer)
```

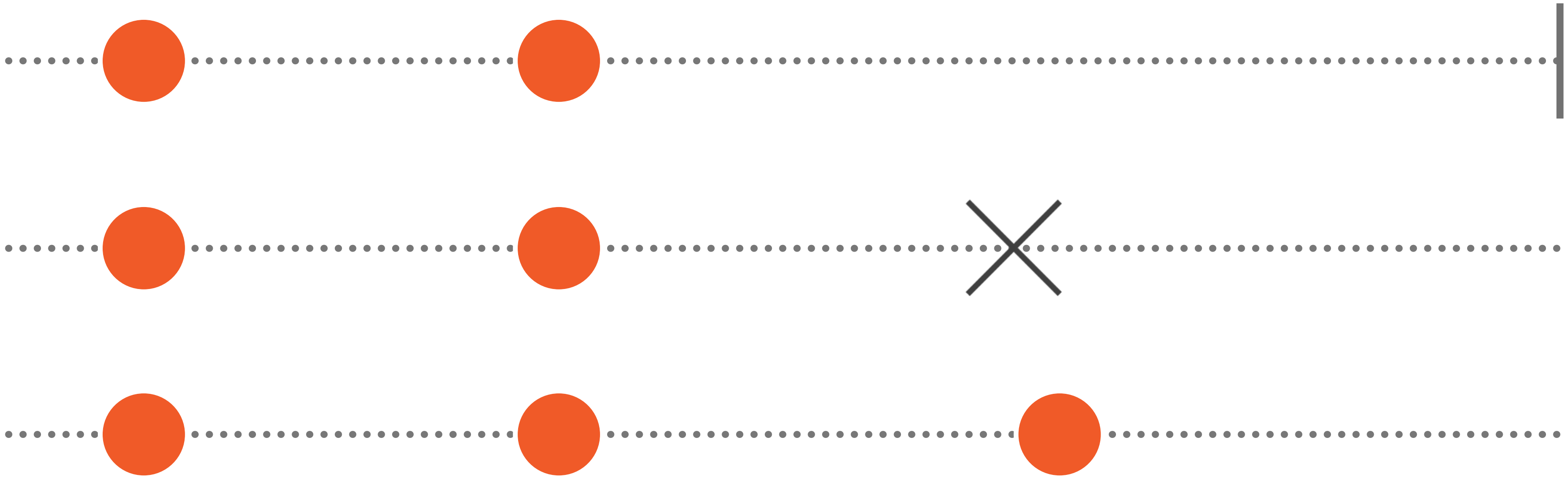
IObserver

```
void OnNext<T>(T value)

void OnCompleted()

void OnError(Exception ex)
```

Marble Diagrams

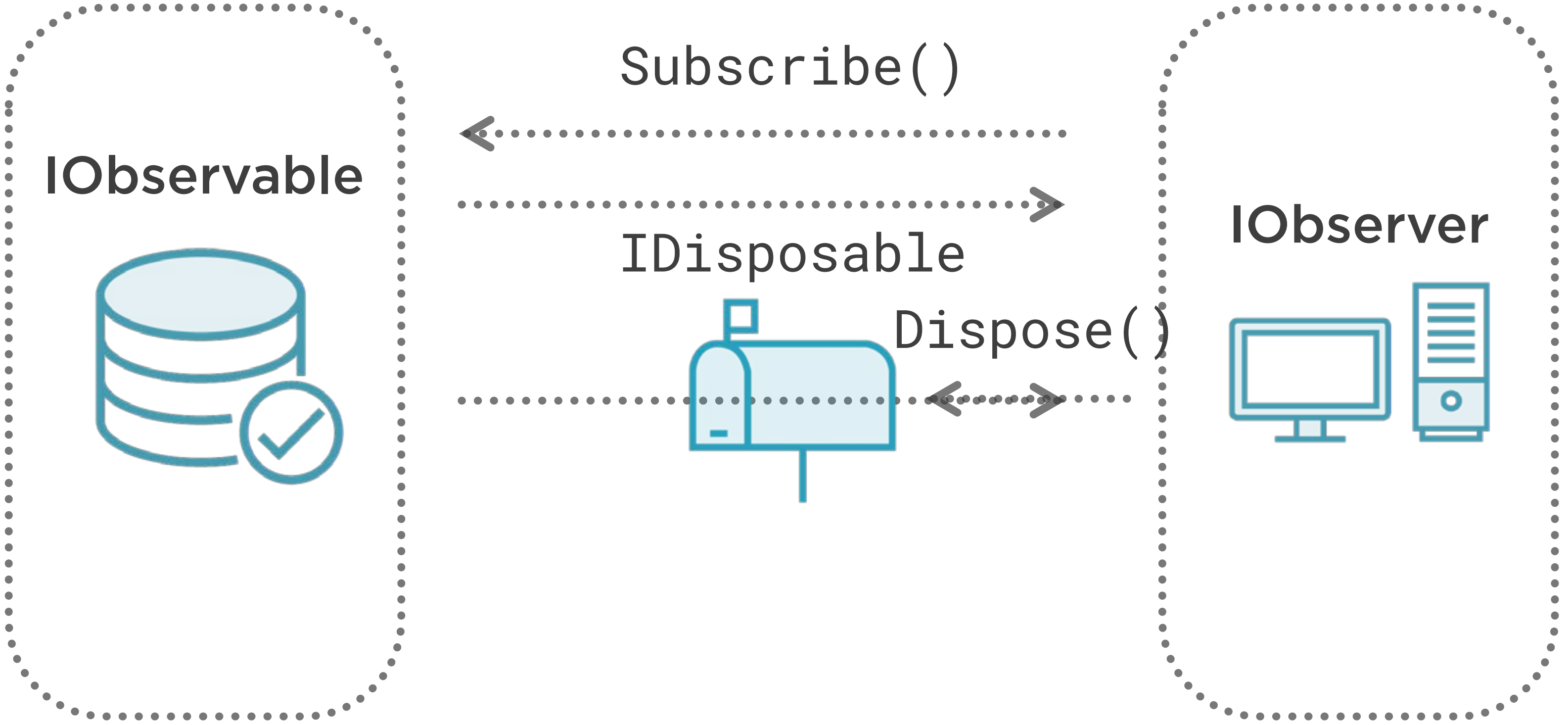


Sample Marble Diagram



OnNext()   OnNext()   OnError()   OnCompleted()

Unsubscribing from IObservable



Demo

Fictitious support ticket management UWP app called “Reactive Tickets”

Implement the search-as-you-type feature

- Using events
- Using observable sequences

Summary

Asynchronous and event-driven programming is difficult

- The code is harder to maintain

Reactive programming treats all changes in data as value sequences in time

- "Events on steroids"

Observers can subscribe to changes in those sequences

Rx adds reactive programming to .NET with full Linq support