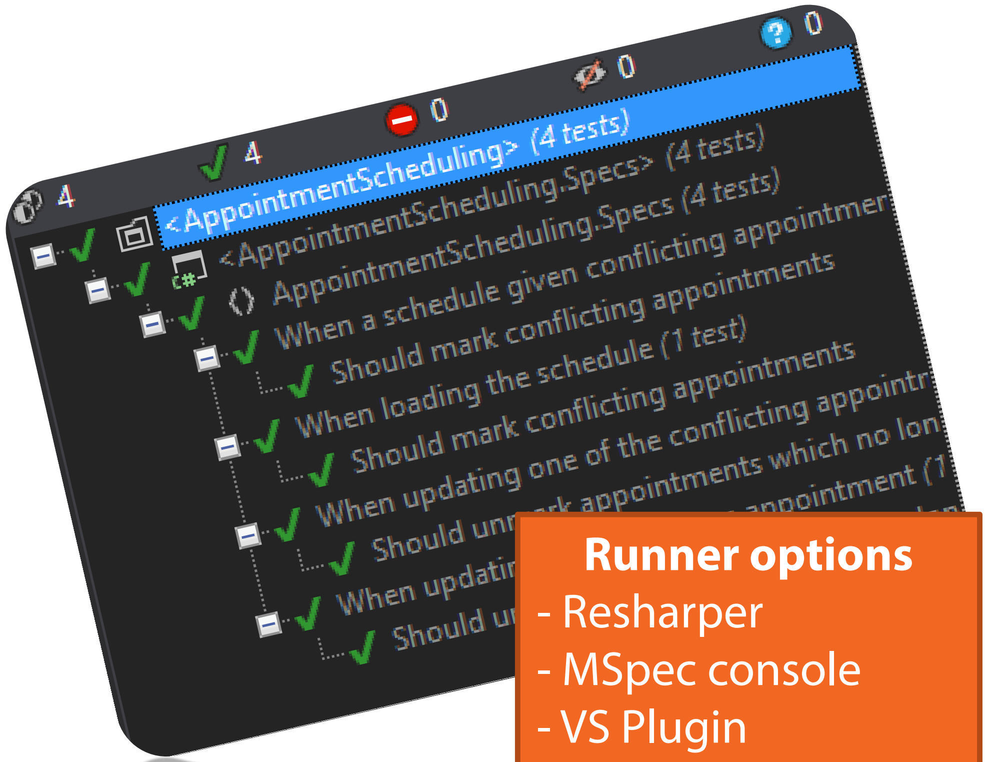
# MSpec Beyond the Basics

Kevin Kuebler

@kevinkuebler

# Where Are We?


4 ✓ 4 0 0 0
<AppointmentScheduling> (4 tests)
<AppointmentScheduling.Specs> (4 tests)
AppointmentScheduling.Specs (4 tests)
When a schedule given conflicting appointmen
Should mark conflicting appointments
When loading the schedule (1 test)
Should mark conflicting appointments
When updating one of the conflicting appointm
Should un rk appointments which no lon
When updatin appointment (1
Should u

**Runner options**
- Resharper
- MSpec console
- VS Plugin

- Three building blocks
  - Establish
  - Because
  - It
- Keeping specs DRY
  - Inheritance
  - Nested contexts

## What’s Next?

- Dealing with exceptions
- Additional attributes
  - More information about context
  - Include/exclude specs
- Output
  - Console runner flags / CI options
- Mocking in MSpec
  - Machine.Fakes

## Expected Exceptions

```
class When_from_account_balance_is_less_than_the_transfer_amount
{
  static Exception ExpectedException;
  Establish context = () => FromAccount = new Account(999);
  Because of = () => ExpectedException = Catch.Exception(
    () => Transfer(FromAccount,ToAccount,1000));
}
```

Wraps the delegate instance provided in a try/catch and returns the exception which is caught

# Expected Exceptions

```
class When_from_account_balance_is_less_than_the_transfer_amount
{
  static Exception ExpectedException;
  Establish context = () => FromAccount = new Account(999);
  Because of = () => ExpectedException = Catch.Exception(
    () => Transfer(FromAccount,ToAccount,1000));
  It Should_not_allow_the_transfer = () =>
    ExpectedException
      .ShouldBeOfExactType<InsufficientFundsException>();
}
```

## Machine.Fakes

- Framework built on top of MSpec
- Provides abstraction over mocking frameworks (provider model), while still allowing direct use of the mocking framework when needed
- Includes an AutoMockingContainer
- WithFakes
  - Base context class which provides abstraction over mocking framework (i.e. "fakes")
- WithSubject<T>
  - Will create instance of T, automatically providing mocks to any constructor parameters that are an interface or abstract base class

## Creating Fakes

- An<T>
  - Abstraction for creating a mock/fake/substitute object

```
var fakeService = An<IAccountService>();
```

- The<T>
  - Abstraction for accessing the automatically created fakes when using WithSubject<T>

```
var fakeService = The<IAccountService>();
```

# Expectations on Fakes

- WhenToldTo(…)
  - Setup an expectation for a fake object

```
var fakeService = The<IAccountService>();
var account = new Account(1000);
fakeService.WhenToldTo(s => s.GetAccount(account.Id)).Return(account);
```

- WasToldTo(…)
  - Verify an expectation on a fake object

```
The<IAccountService>().WasToldTo(s => s.SuspendAccount(account));
```

## Expectations on Fakes

- Param, Param<T>
  - Provides multiple ways to verify parameters in WasToldTo() calls (can also be used to setup different expectations in WhenToldTo() calls)

```
.WasToldTo(s => s.SuspendAccount(Param.Is(account)));
```

```
.WasToldTo(s => s.SuspendAccount(Param.IsAny<Account>()));
```

```
.WasToldTo(s => s.SuspendAccount(Param<Account>.Matches(a => a.Id == account.Id)));
```

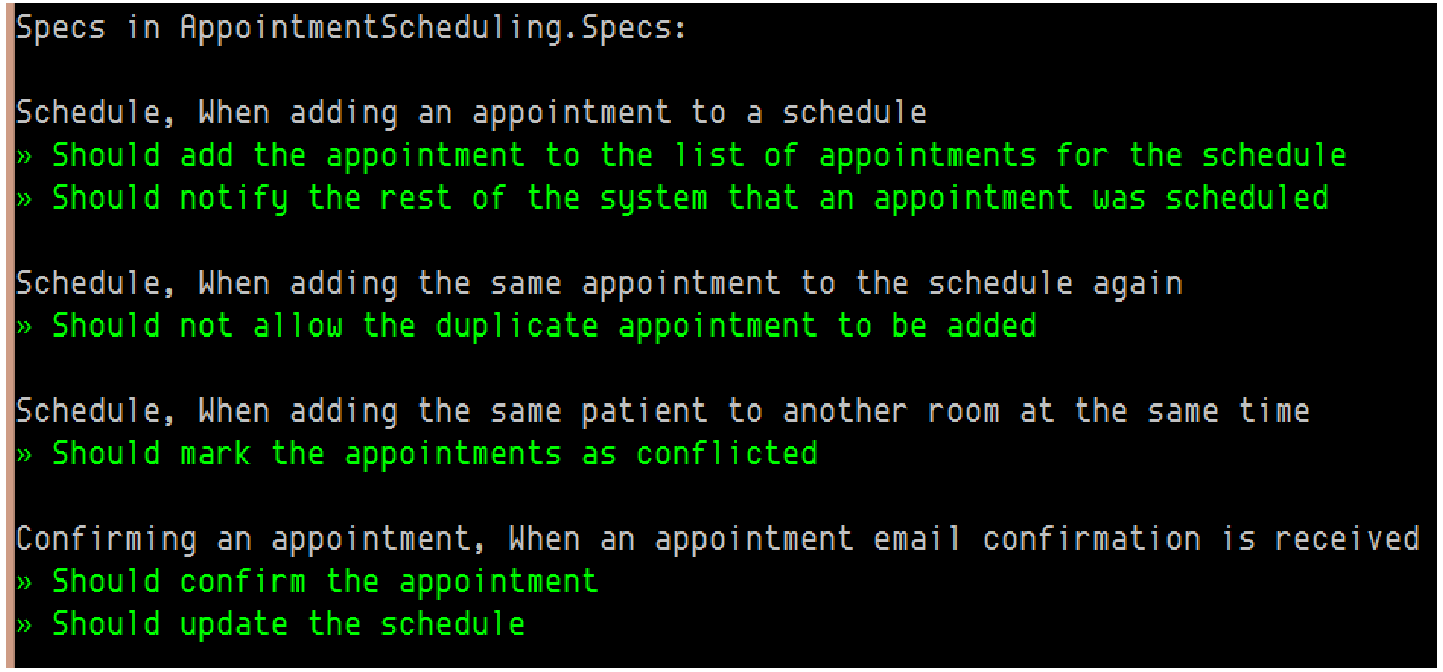
```
.WasToldTo(s => s.SuspendAccount(Param<Account>.IsNotNull));
```

# Summary and Final Thoughts

- Expected Exceptions
  - Catch.Exception()
- [Subject]
  - Allows for more clearly defined and categorized spec results
- [Tag]
  - Provides categories to test runner for including/excluding groups of specs

# Summary and Final Thoughts

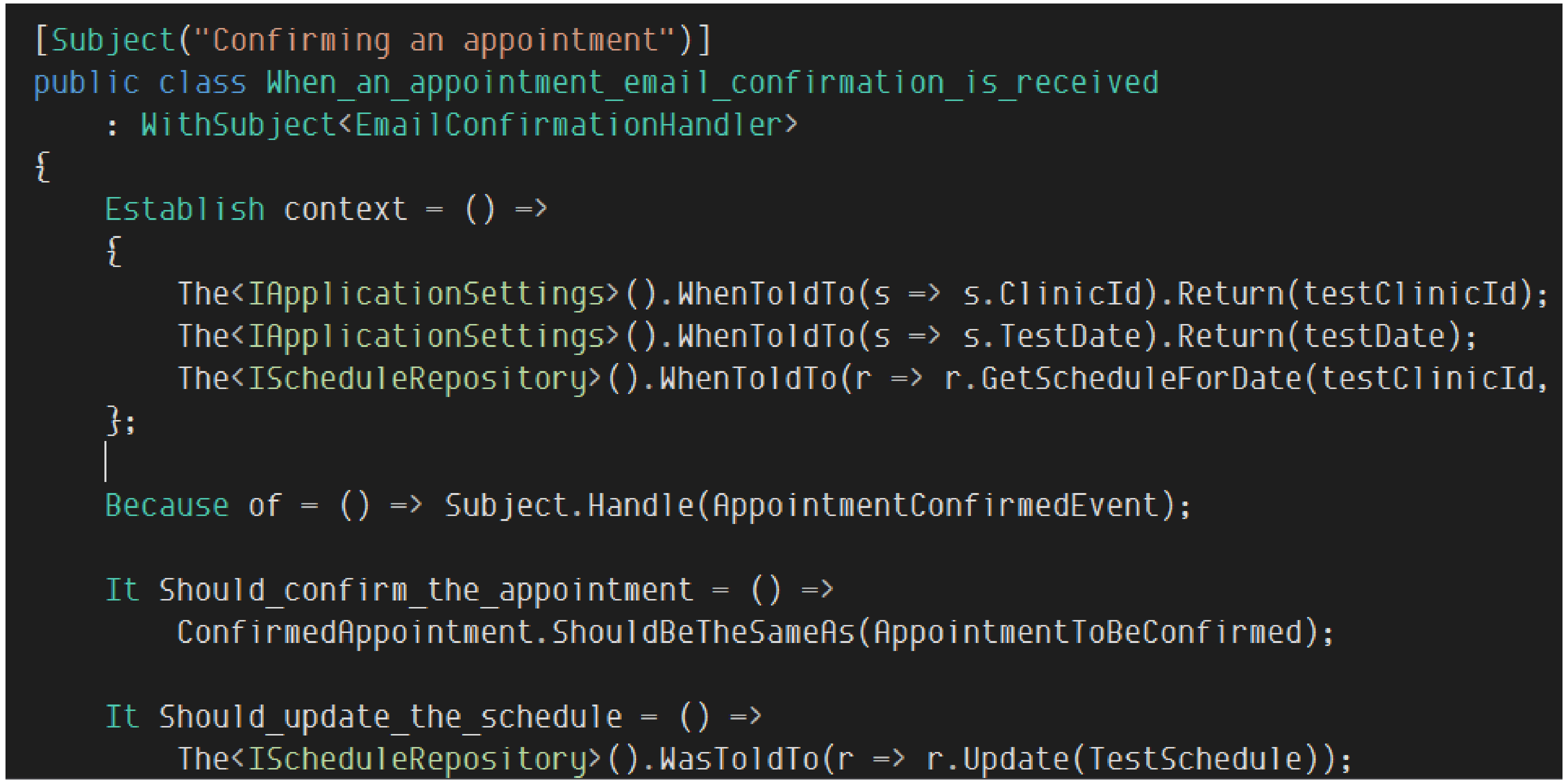
- Console Runner
  - Include/Exclude with Tags
  - HTML results
  - XML results
  - CI integration
    - TeamCity
    - AppVeyor



```
Specs in AppointmentScheduling.Specs:

Schedule, When adding an appointment to a schedule
» Should add the appointment to the list of appointments for the schedule
» Should notify the rest of the system that an appointment was scheduled

Schedule, When adding the same appointment to the schedule again
» Should not allow the duplicate appointment to be added

Schedule, When adding the same patient to another room at the same time
» Should mark the appointments as conflicted

Confirming an appointment, When an appointment email confirmation is received
» Should confirm the appointment
» Should update the schedule
```

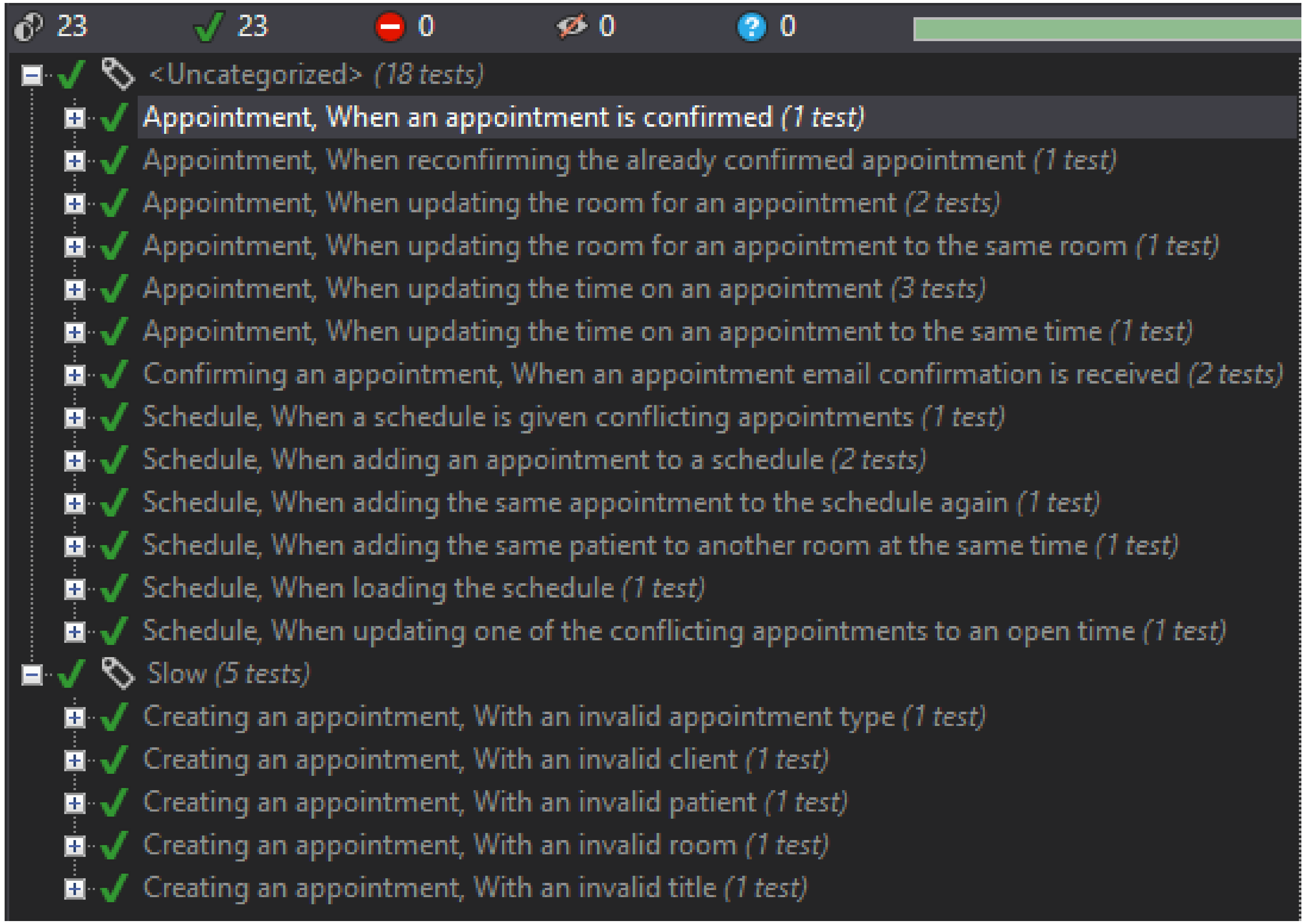
## Summary and Final Thoughts

- Machine.Fakes
  - Built on MSpec and existing mocking frameworks
  - Mocking abstractions and auto-mocking container
  - Easy, clean integration with MSpec



```
[Subject("Confirming an appointment")]
public class When_an_appointment_email_confirmation_is_received
    : WithSubject<EmailConfirmationHandler>
{
    Establish context = () =>
    {
        The<IApplicationSettings>().WhenToldTo(s => s.ClinicId).Return(testClinicId);
        The<IApplicationSettings>().WhenToldTo(s => s.TestDate).Return(testDate);
        The<IScheduleRepository>().WhenToldTo(r => r.GetScheduleForDate(testClinicId,
    };

    Because of = () => Subject.Handle(AppointmentConfirmedEvent);

    It Should_confirm_the_appointment = () =>
        ConfirmedAppointment.ShouldBeTheSameAs(AppointmentToBeConfirmed);

    It Should_update_the_schedule = () =>
        The<IScheduleRepository>().WasToldTo(r => r.Update(TestSchedule));
```

# Summary and Final Thoughts



```
23   ✓ 23   0   0   0
<Uncategorized> (18 tests)
  Appointment, When an appointment is confirmed (1 test)
  Appointment, When reconfirming the already confirmed appointment (1 test)
  Appointment, When updating the room for an appointment (2 tests)
  Appointment, When updating the room for an appointment to the same room (1 test)
  Appointment, When updating the time on an appointment (3 tests)
  Appointment, When updating the time on an appointment to the same time (1 test)
  Confirming an appointment, When an appointment email confirmation is received (2 tests)
  Schedule, When a schedule is given conflicting appointments (1 test)
  Schedule, When adding an appointment to a schedule (2 tests)
  Schedule, When adding the same appointment to the schedule again (1 test)
  Schedule, When adding the same patient to another room at the same time (1 test)
  Schedule, When loading the schedule (1 test)
  Schedule, When updating one of the conflicting appointments to an open time (1 test)
Slow (5 tests)
  Creating an appointment, With an invalid appointment type (1 test)
  Creating an appointment, With an invalid client (1 test)
  Creating an appointment, With an invalid patient (1 test)
  Creating an appointment, With an invalid room (1 test)
  Creating an appointment, With an invalid title (1 test)
```

## References and Other Courses

- *Domain-Driven Design Fundamentals* (Smith, Lerman)
- *Pragmatic Behavior-driven Design with .NET* (Conery)
- *Automated Testing: End to End* (Roberts)
- *Approval Tests for .NET* (Roberts)
- Machine.Specifications and Machine.Fakes on Github
  - https://github.com/machine/machine.specifications
  - https://github.com/machine/machine.fakes