# Boosting Test Speed with Parallel Testing and Targeted Automation


André Koene

newspark
a BlueGroup IT company

TEST AUTOMATION DAYS 2025
INITIATED AND ORGANIZED BY CKC SEMINARS

A significant portion of the images used in this presentation were generated using the AI service ChatGPT. These images were created based on descriptions provided by the presenter and are intended for illustrative purposes only.

# Me

André Koene

sogeti Part of Capgemini

newspark a BlueGroup IT company

FYNCH EVERY TRIP COUNTS

kpn

Rabobank

VEON

roxit

vereniging eigen huis

sta sterker














# Me

André Koene


# 🧭 Navigation

- 🔍 **What**
- ❓ **Why**
- 📜 **How**

- 🎯 **T** argeted
- 🛡️ **R** eliable
- 💡 **I** nformative
- 🔧 **M** aintainable
- ⚡ **S** peedy

# 🔍 What

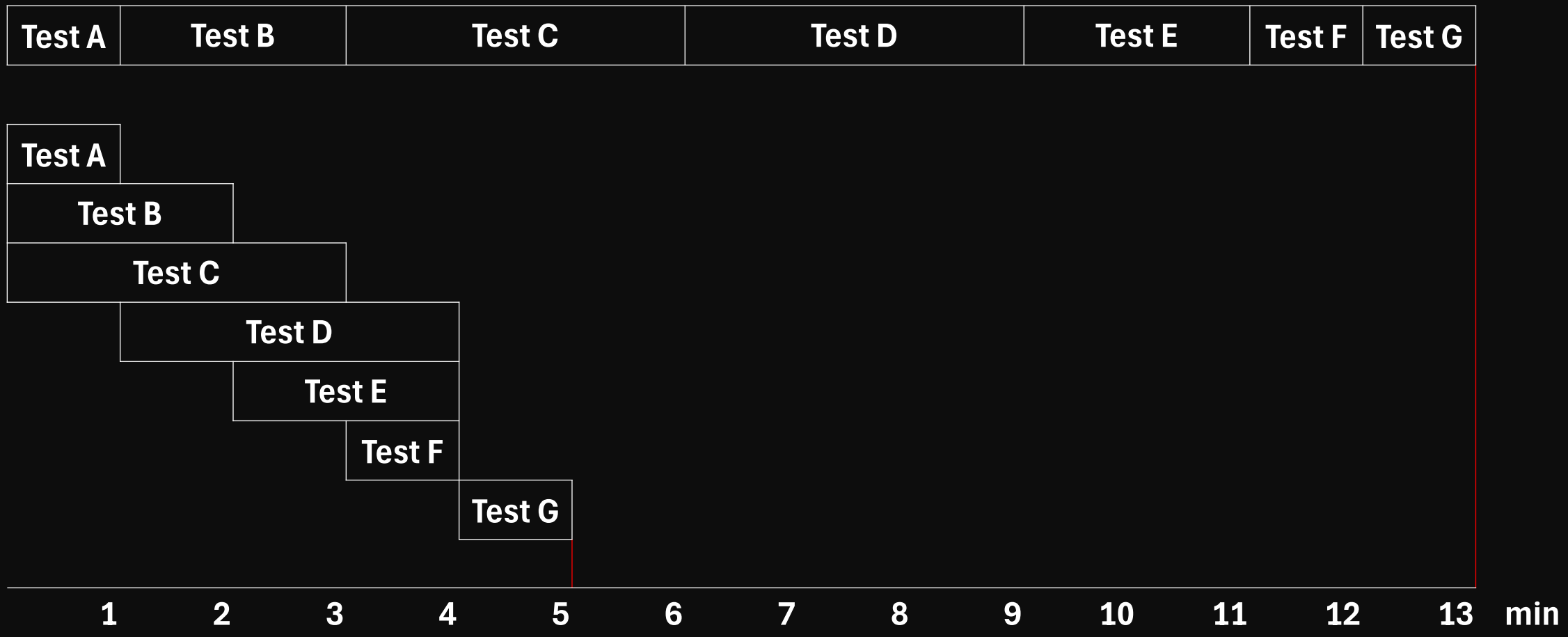
“Parallel testing is a method used to run multiple **automated tests** at the same time, rather than one after another (sequentially).”


**Order process** Endpoints related to the order process

| | | |
|---|---|---|
| GET | /cart/{customerId} | Get the shopping cart contents |
| POST | /cart/{customerId}/addItem | Add an item to the shopping cart |
| POST | /cart/{customerId}/placeOrder | Place an order |

**Customer data** Endpoints related to the customer data

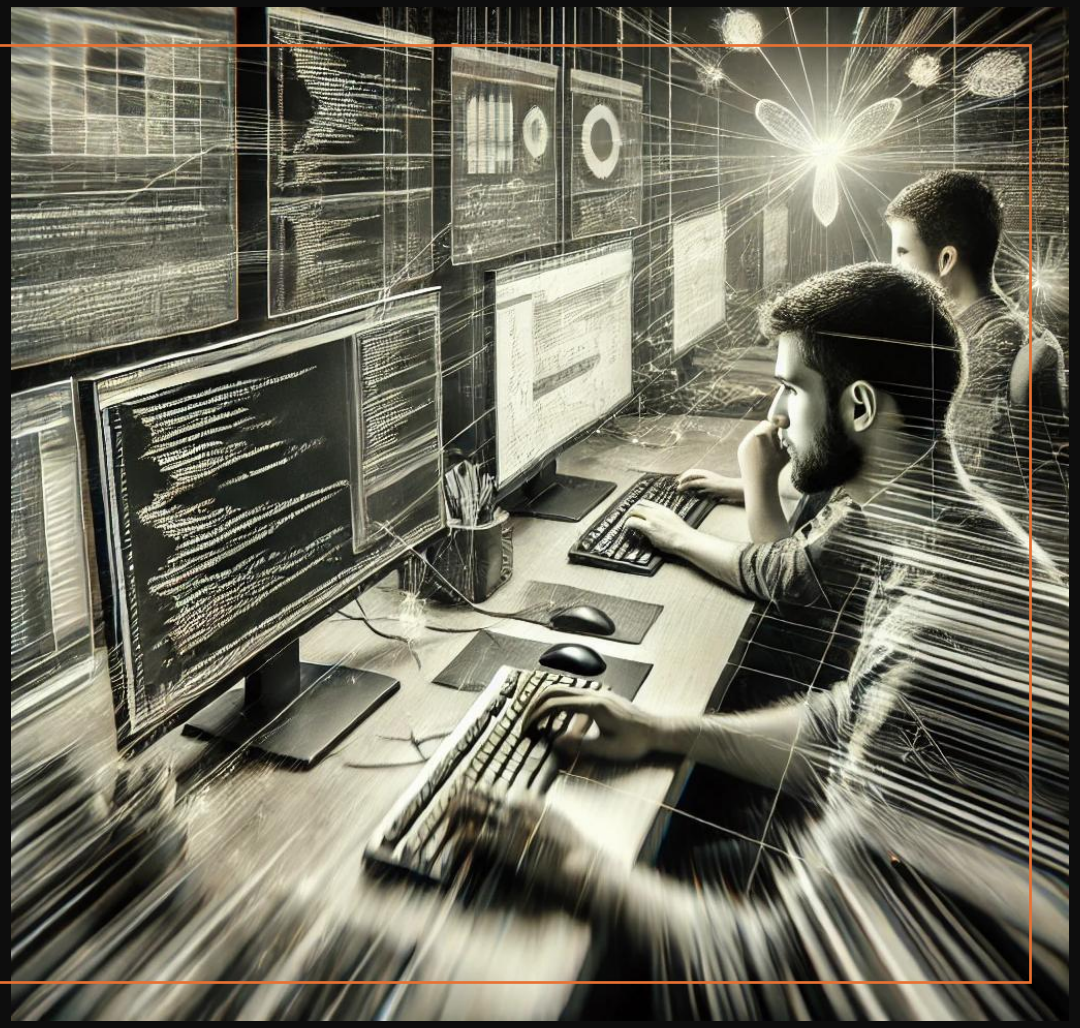
| | | |
|---|---|---|
| GET | /orders/{customerId}/history | Get customer order history |


Test A
Test B
Test C
Test D
Test E
Test F
Test G
Test A
Test B
Test C
Test D
Test E
Test F
Test G
1
2
3
4
5
6
7
8
9
10
11
12
13
min

# ❓ Why

**Context switches are killing**

“Fast feedback from tests enables quicker bug fixes and immediate retesting while the details are still fresh in mind.”



# How

github.com/AndreD3v/ReqNRollerCoaster

docs.reqnroll.net/latest/execution/parallel-execution.html

cucumber.io/docs/guides/parallel-execution

docs.robotframework.org/docs/parallel

# 🎓 TRIMS

“TRIMS is a mnemonic for creating valuable automation that truly supports your testing.”

**Richard Bradshaw**

# 🎓 TRIMS

“Every principle plays a crucial role in accelerating test feedback”

🎯 **T** argeted

🛡️ **R** eliable

💡 **I** nformative

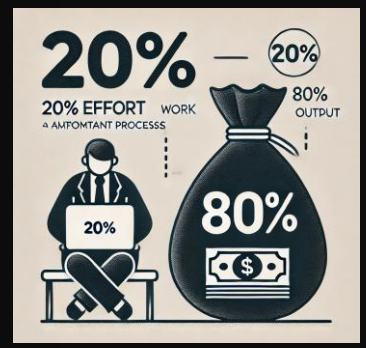
🔧 **M** aintainable

⚡ **S** peedy

# 🎯 Targeted


**Risk**
**=**
**Probability**
**x**
**Severity**

```
Scenario: combine Scenarios A and B
    Given I have a default customer
    When I do something
    Then I should see 'XXX'
    When I do something else
    Then I should see 'YYY'
```




# Reliable

TRIMS

```
4
5   Scenario: Test A
6       Given I have a default customer
7       When I do something
8       Then I should see a customer with property 'XXX'
9
10  Scenario: Test B
11      Given I have a default customer
12      When I do something else
13      Then I should see a customer with property 'YYY'
```

log_20250124152515692025012 4.log

```
1  [15:25:03.751+01:00 INF] Test A | customerId= '7777'
2  [15:26:05.754+01:00 INF] Test A | customerId= '7777' has done something
3  [15:27:07.765+01:00 INF] Test B | customerId= '8888'
4  [15:28:10.766+01:00 INF] Test B | customerId= '8888' has done something else
5  [15:29:12.757+01:00 INF] Test A | customerId= '7777' has 'XXX', while expected 'XXX'
6  [15:30:19.766+01:00 INF] Test B | customerId= '7777' has 'XXX', while expected 'YYY'
7
```

# Context leakage

```
private int _executedTestScenarios = 0;

[Before Scenario]                    [BeforeTestRun]
public void BeforeScenario()
{
        // Only execute this block on the very first test scenario
        if (_executed stScenarios == 0)
        {
            Task.Run(() => CleanupOldData()).Wait();
        }
        else
        {
            Log.Information($"Skipped testdata clean-up for this individual scenario");
        }
        _executedTe cenarios++
}
```

Deadlocks

# Informative

Test run finished: 9 Tests (8 Passed, 1 Failed, 0 Skipped) run in 12,7 sec

TRIMS

Test

- ReqNRollerCoasterProject (9)
  - ReqNRollerCoasterProject.Fe...
    - AddNumbersFeature (3)
      - AddANegativeNumbers
      - AddTwoNumbers
      - AddTwoOtherNumbers
    - DivideNumbersFeature (3)
    - MultiplyNumbersFeature (3)

Run | Debug | Ask Copilot

**Test Detail Summary**

AddTwoNumbers

Source: AddNumbers.feature line 9

Duration: 4,1 sec

Message:

```
Expected _result to be 120, but found 130 (difference of 10).
```

Stack Trace:

Standard Output:

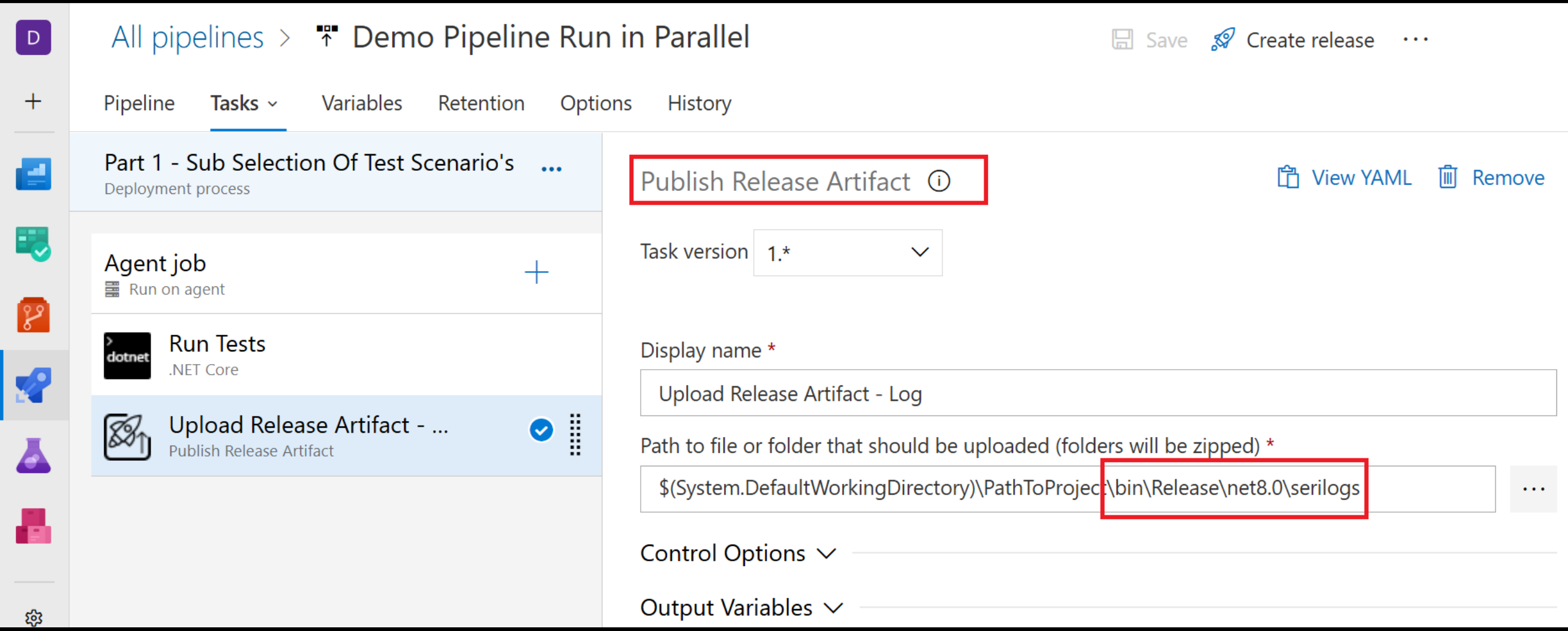
```
[19:54:49.318+01:00 INF] BeforeScenario - Starting scenario: Add two numbers
Given the first number is 50
[19:54:49.318+01:00 INF] _firstNumber = 50
-> done: CalculatorStepDefinitions.GivenTheFirstNumberIs(50) (0,0s)
And the second number is 70
[19:54:50.332+01:00 INF] _secondNumber = 70
-> done: CalculatorStepDefinitions.GivenTheSecondNumberIs(70) (0,0s)
When the two numbers are added
[19:54:51.344+01:00 INF] _result = 130
-> done: CalculatorStepDefinitions.WhenTheTwoNumbersAreAdded() (0,0s)
Then the result should be 120
-> error: Expected _result to be 120, but found 130 (difference of 10). (0,1s)
```

# Clear and organized logging

log_20250128195445192025012 8.log

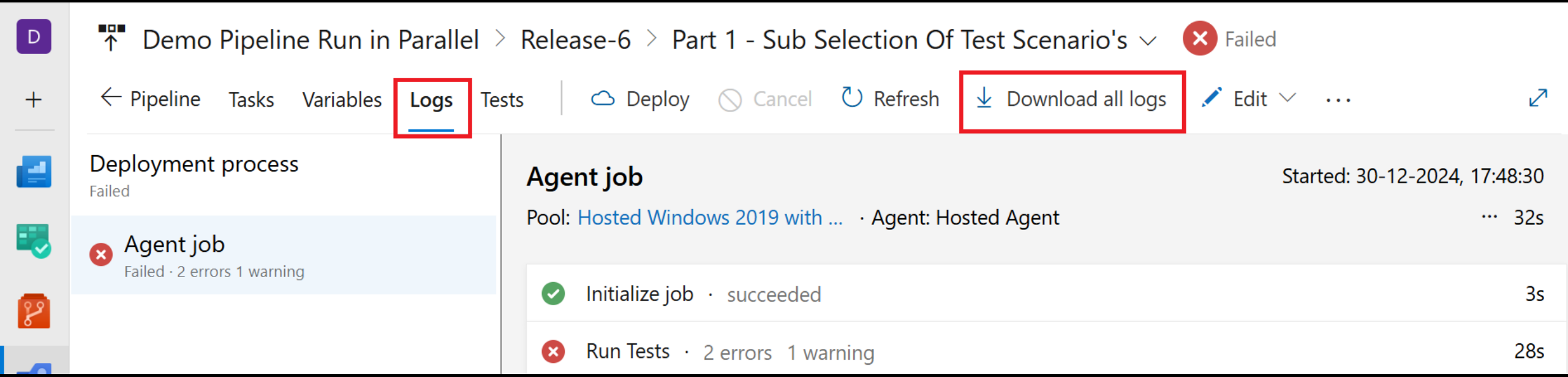
```
13 [19:54:47.283+01:00 INF] Add a negative numbers | _result = 46
14 [19:54:49.318+01:00 INF] Divide two numbers | BeforeScenario - Starting scenario: Divide two numbers
15 [19:54:49.318+01:00 INF] Add two numbers | BeforeScenario - Starting scenario: Add two numbers
16 [19:54:49.318+01:00 INF] Multiply two numbers | BeforeScenario - Starting scenario: Multiply two numbers
17 [19:54:49.318+01:00 INF] Divide two numbers | firstNumber = 100
18 [19:54:49.318+01:00 INF] Add two numbers | firstNumber = 50
19 [19:54:49.318+01:00 INF] Multiply two numbers | firstNumber = 60
20 [19:54:50.332+01:00 INF] Divide two numbers | secondNumber = 20
21 [19:54:50.332+01:00 INF] Add two numbers | secondNumber = 70
22 [19:54:50.332+01:00 INF] Multiply two numbers | _secondNumber = 70
23 [19:54:51.344+01:00 INF] Multiply two numbers | _result = 4200
24 [19:54:51.344+01:00 INF] Divide two numbers | result = 5
25 [19:54:51.344+01:00 INF] Add two numbers | result = 130
26 [19:54:53.365+01:00 INF] Divide two other numbers | BeforeScenario - Starting scenario: Divide two other
27 [19:54:53.365+01:00 INF] Divide two other numbers | _firstNumber = 60
```

# 💡 Clear and organized logging




# logging in the pipeline

TRIMS

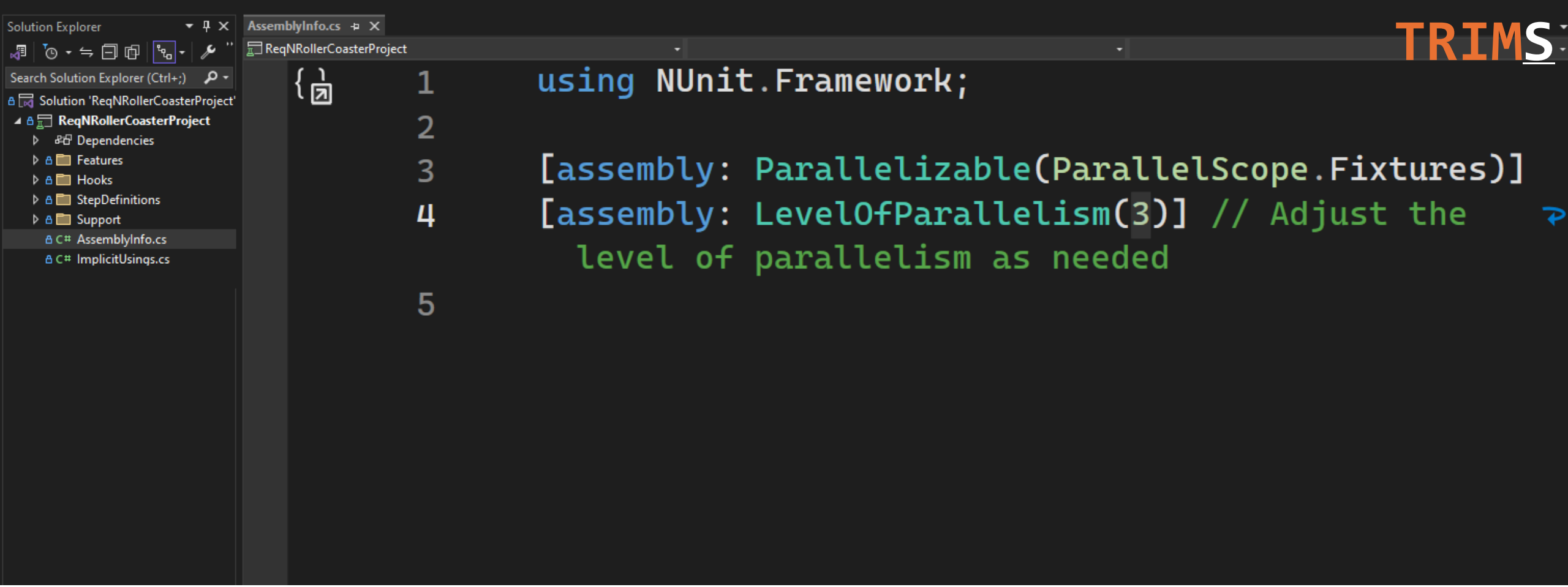



# logging in the pipeline

# 🔧 Maintainable



# Speedy




TRIMS

# Greenfield setup

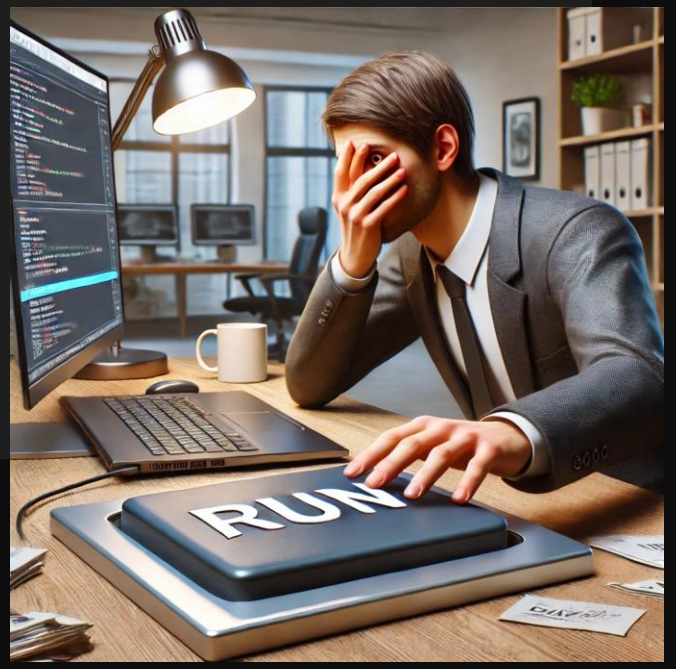

Given/When

# TRIMS

AssemblyInfo.cs

ReqNRollerCoasterProject

```
1  using NUnit.Framework;
2
3  [assembly: Parallelizable(ParallelScope.Fixtures)]
4  [assembly: LevelOfParallelism(3)] // Adjust the
     level of parallelism as needed
5
```




#@##
@#??
STUCK
02.39.23
LOGS

```
/// <summary>
/// Cleanup statics from 'Context' class before each testscenario
/// </summary>
[BeforeScenario]
0 references
public void BeforeScenarioCleanupStatics()
{
    // Add 'scenario' scope to seri log
    LogContext.PushProperty("Scenario", _scenarioContext.ScenarioInfo.Title);

    string[] assignedTags = _scenarioContext.ScenarioInfo.CombinedTags;
    if (assignedTags.Contains("RunInParallel"))
    {
        Log.Information("BeforeScenario - This test scenario should be able to run in parallel");
    }
    else
    {
        Log.Information("All shared 'Context' objects will be cleared");
        //todo: these contexts must be phased out, to run in parallel: It should not be necessary to reset
          contexts between multiple scenarios:
        Context.DummyA = null;
        Context.DummyB = null;
        Context.DummyC = null;
        Context.DummyD = null;
        Context.DummyE = null;
    }
}
```

# TRIMS

log_20250124152515692025012 4.log

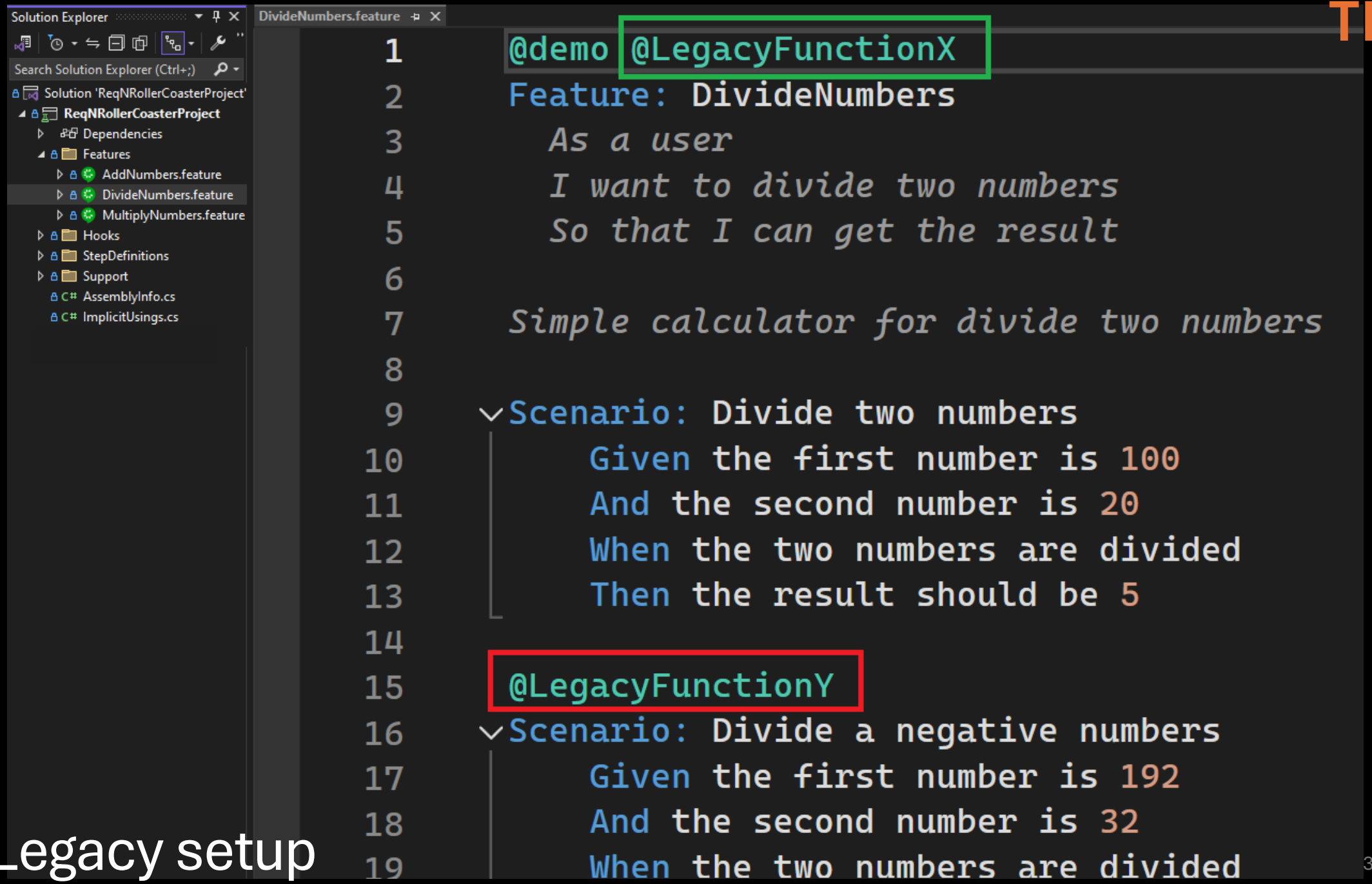
```
1  [15:25:03.751+01:00 INF] Test A | customerId= '7777'
2  [15:26:05.754+01:00 INF] Test A | customerId= '7777' has done something
3  [15:27:07.765+01:00 INF] Test B | customerId= '8888'
4  [15:28:10.766+01:00 INF] Test B | customerId= '8888' has done something else
5  [15:29:12.757+01:00 INF] Test A | customerId= '7777' has 'XXX', while expected 'XXX'
6  [15:30:19.766+01:00 INF] Test B | customerId= '7777' has 'XXX', while expected 'YYY'
7
```

AssemblyInfo.cs

ReqNRollerCoasterProject

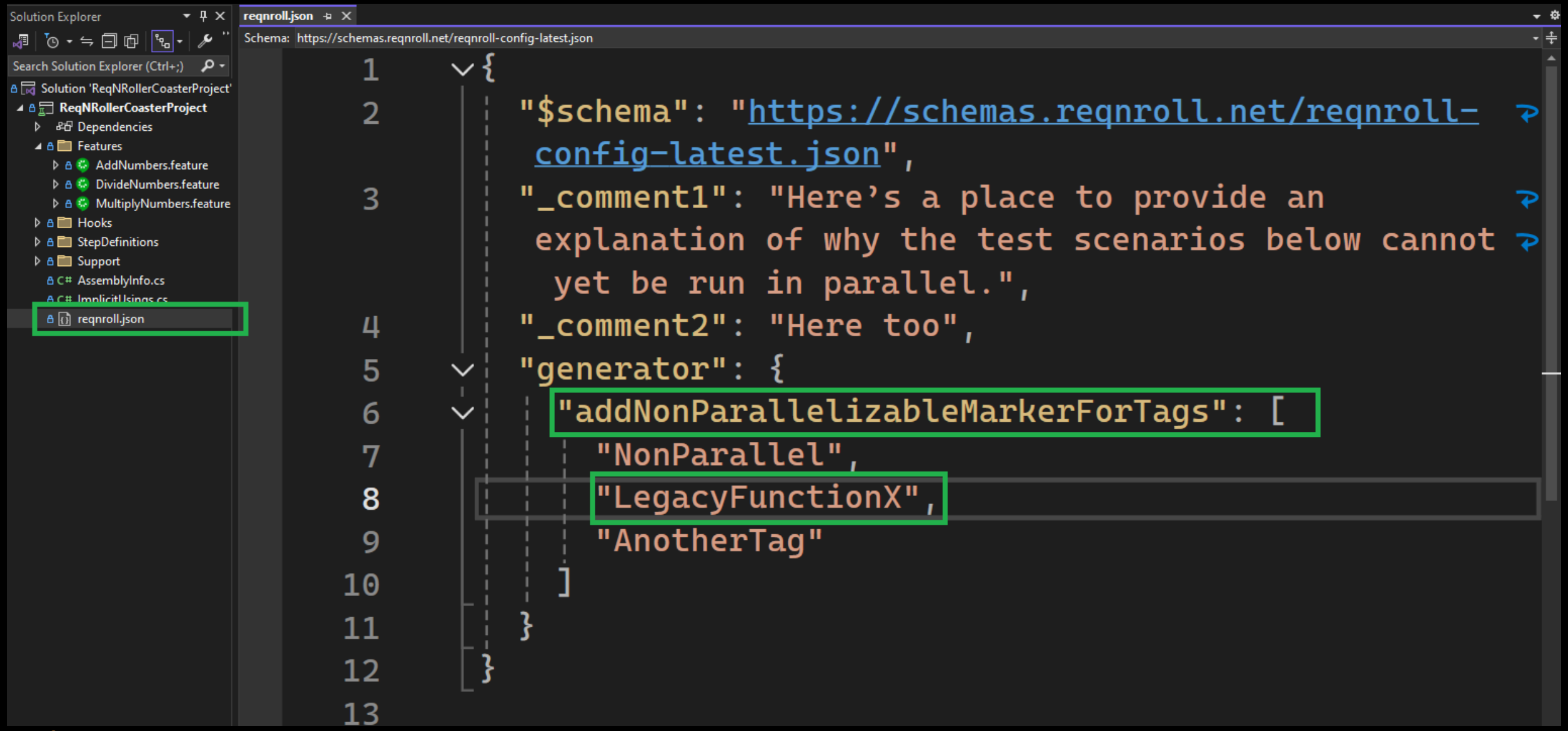
```
1  using NUnit.Framework;
2
3  [assembly: Parallelizable(ParallelScope.Fixtures)]
4  [assembly: LevelOfParallelism(3)] // Adjust the
     level of parallelism as needed
5
```

# Legacy setup

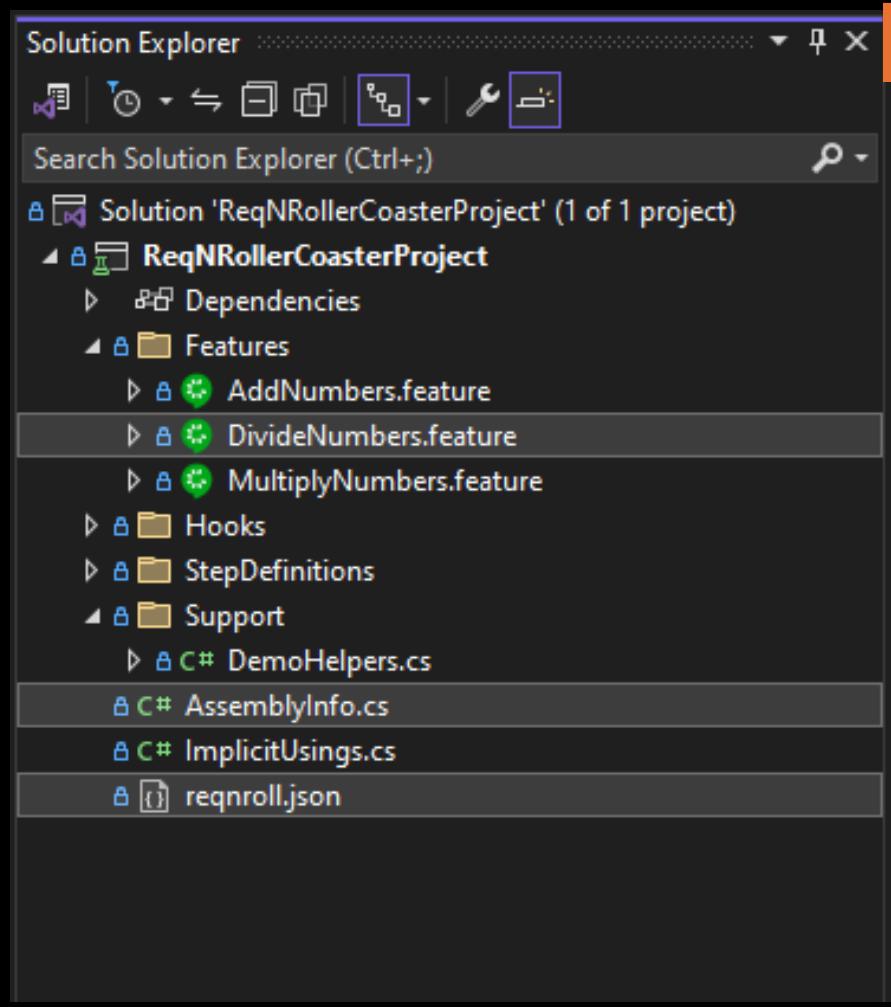



TRIMS

Legacy setup




# Legacy setup




AssemblyInfo.cs | DivideNumbers.feature | reqnroll.json

```
1   @demo @LegacyFunctionX
2   Feature: DivideNumbers
3     As a user
4     I want to divide two numbers
5     So that I can get the result
6
7   Simple calculator for divide two numbers
8
9   Scenario: Divide two numbers
10      Given the first number is 100
11      And the second number is 20
12      When the two numbers are divided
13      Then the result should be 5
14
```

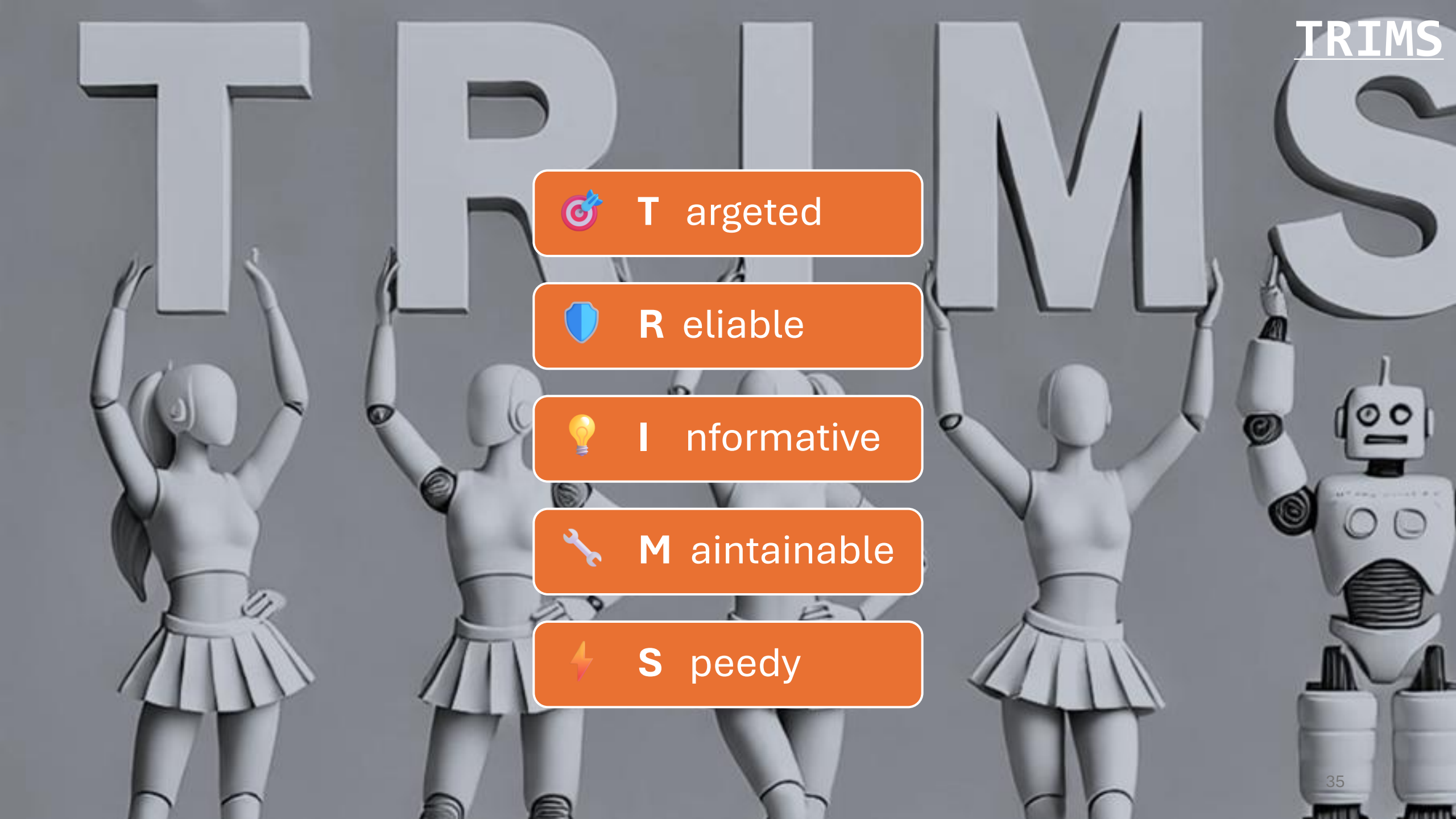
# Legacy setup

TRIMS

```
/// <summary>
/// Cleanup statics from 'Context' class before each testscenario
/// </summary>
[BeforeScenario]
0 references
public void BeforeScenarioCleanupStatics()
{
    // Add 'scenario' scope to seri log
    LogContext.PushProperty("Scenario", _scenarioContext.ScenarioInfo.Title);

    string[] assignedTags = _scenarioContext.ScenarioInfo.CombinedTags;
    if (assignedTags.Contains("RunInParallel"))
    {
        Log.Information("BeforeScenario - This test scenario should be able to run in parallel");
    }
    else
    {
        Log.Information("All shared 'Context' objects will be cleared");
        //todo: these contexts must be phased out, to run in parallel: It should not be necessary to reset
          contexts between multiple scenarios:
        Context.DummyA = null;
        Context.DummyB = null;
        Context.DummyC = null;
        Context.DummyD = null;
        Context.DummyE = null;
    }
}
```

Legacy setup




- 🎯 **T** argeted
- 🛡️ **R** eliable
- 💡 **I** nformative
- 🔧 **M** aintainable
- ⚡ **S** peedy

# To summarize

**Greenfield**: Start parallel testing early—initial setup is simple.

**Implementation:** Use scenario tags, JSON config, and custom code if needed.

**Legacy:** Use a phased approach to introduce parallel runs.

**Both:** Clear custom logging is essential for quick issue tracing.

newspark
a BlueGroup IT company

Give yourself access to all the content and url's from this presentation. Start boosting your test speed tomorrow!

TEST AUTOMATION DAYS 2025
INITIATED AND ORGANIZED BY CKC SEMINARS