

C#  
this.\_course.Dispose();

Rasmus Lystrøm  
Associate Professor  
ITU  
[rne@itu.dk](mailto:rne@itu.dk)

The screenshot shows the Visual Studio Code editor with two C# files open: ProgramTests.cs and Program.cs. The left sidebar contains the standard VS Code toolbars. The bottom of the editor shows a terminal window with the output of a dotnet test command.

```
ProgramTests.cs
1 using System;
2 using System.IO;
3 using Xunit;
4
5 namespace HelloWorld.Tests
6 {
7     0 references | Run All Tests | Debug All Tests
8     public class ProgramTests
9     {
10         [Fact]
11         0 references | Run Test | Debug Test
12         public void Main_given_no_args_p
13         {
14             // Arrange
15             using var writer = new StringWriter();
16             Console.SetOut(writer);
17
18             // Act
19             Program.Main(new string[0]);
20
21             // Assert
22             var output = writer.GetStringBuilder().ToString();
23             Assert.Equal("Hello World!", output);
24         }
25     }
26 }
```

```
Program.cs
1 using System;
2
3 namespace HelloWorld
4 {
5     1 reference
6     public class Program
7     {
8         1 reference
9         public static void Main(string[] args)
10         {
11             Console.WriteLine("Hello World!");
12         }
13     }
14 }
```

Terminal Output:

```
1: pwsh
Loading personal and system profiles took 1008ms.
C:\HelloWorld> dotnet test
Test run for C:\HelloWorld\HelloWorld.Tests\bin\Debug\netcoreapp3.0\HelloWorld.Tests.dll (.NETCoreApp, Version=v3.0)
Microsoft (R) Test Execution Command Line Tool Version 16.3.0
Copyright (c) Microsoft Corporation. All rights reserved.

Starting test execution, please wait...

A total of 1 test files matched the specified pattern.

Test Run Successful.
Total tests: 1
Passed: 1
Total time: 3,4618 Seconds
C:\HelloWorld>
```

# Agenda

Security Recap

DevOps

Exam questions

# Security Recap

**When in doubt; read  
the docs!**

Source: <http://keepcalm-o-matic.co.uk/p/keep-calm-and-rtfm-123>



**KEEP  
CALM  
AND**

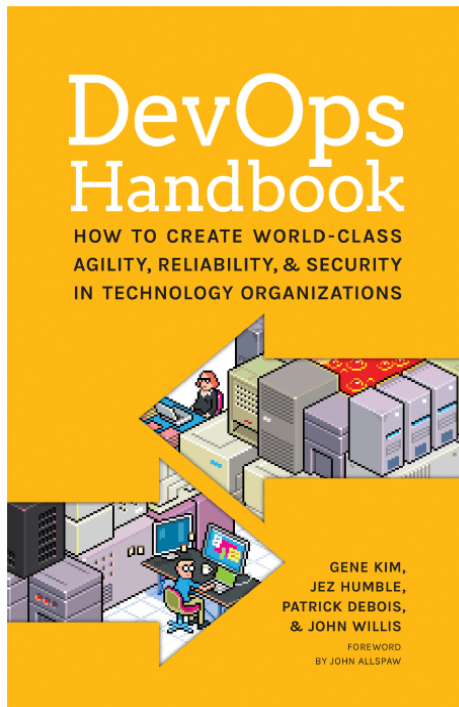
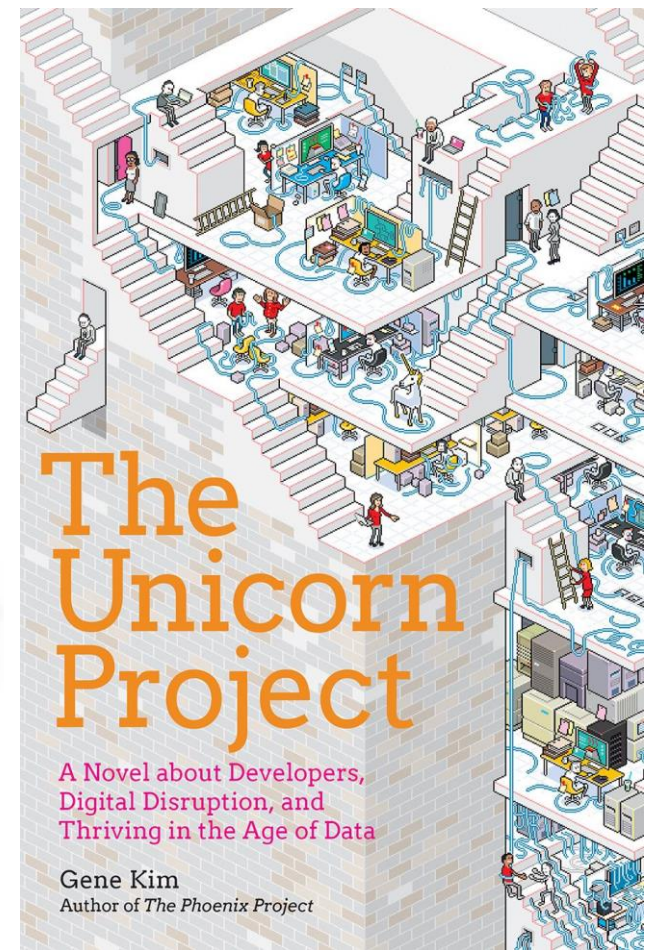
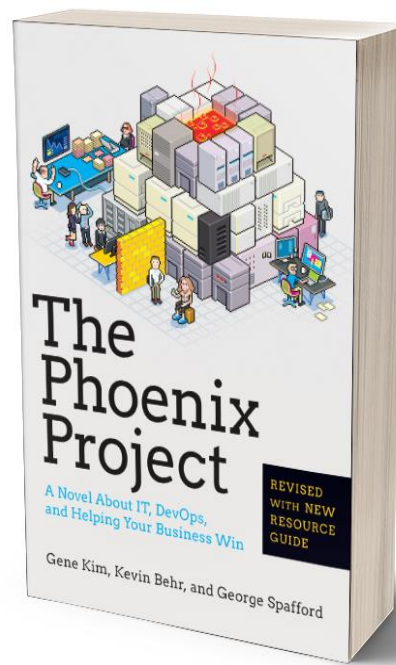
**RTFM**

DevOps

# DevOps

**DevOps** is a set of practices that automates the processes between software development and IT teams, in order that they can build, test, and release software faster and more reliably. The concept of **DevOps** is founded on building a culture of collaboration between teams that historically functioned in relative siloes.

# Books



# Continuous Integration



# Continuous Integration

**Continuous integration** (CI) is the practice of automating the **integration** of code changes from multiple contributors into a single software project. The CI process is comprised of automatic tools that assert the new code's correctness before **integration**.

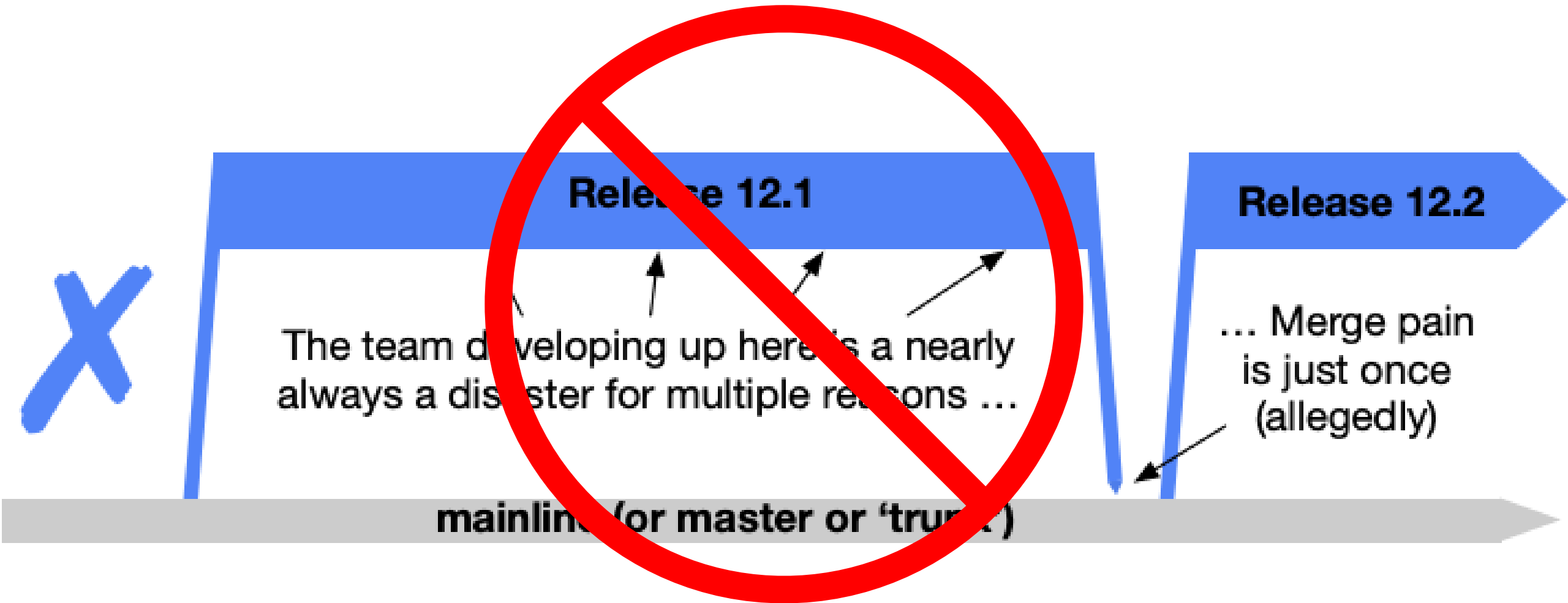
# Trunk Based Development

# Trunk Based Development

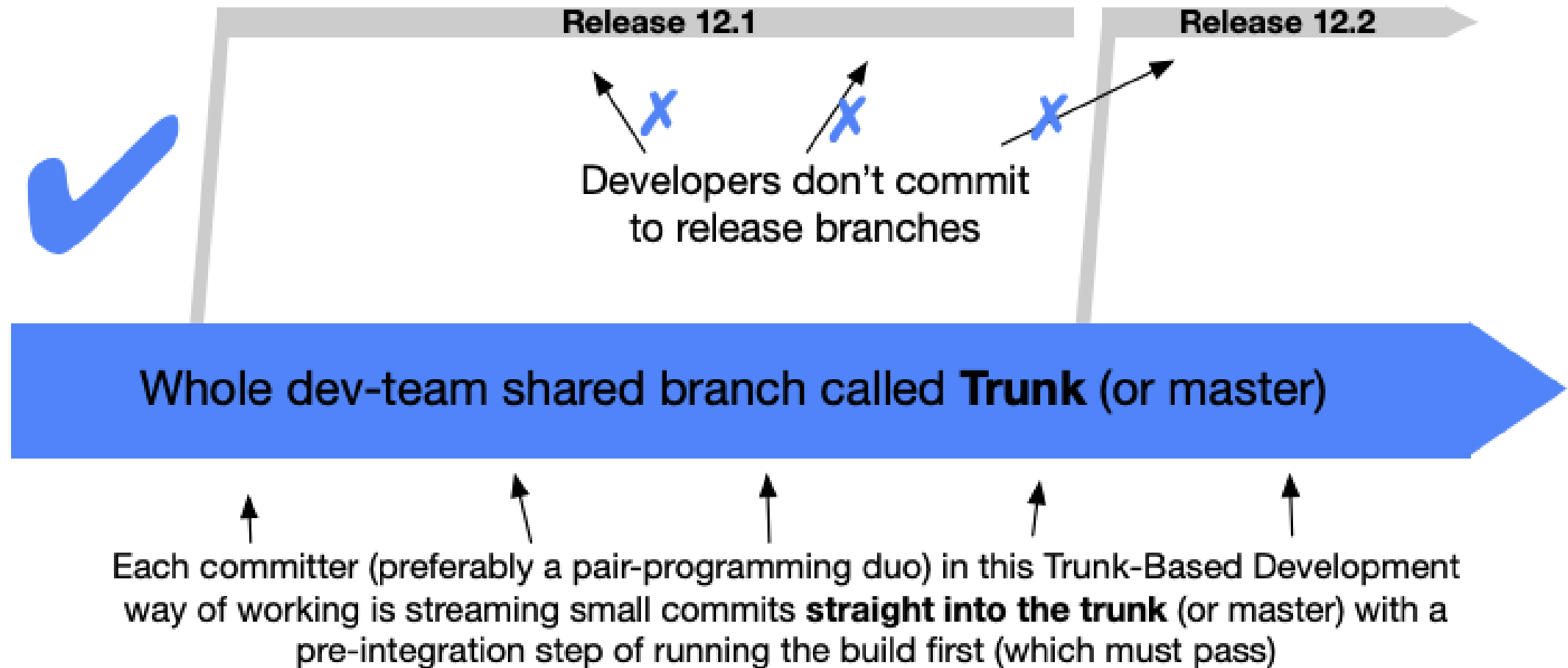
A source-control branching model, where developers collaborate on code in a single branch called 'trunk' \*, resist any pressure to create other long-lived development branches by employing documented techniques. They therefore avoid merge hell, do not break the build, and live happily ever after.

\* master, in Git nomenclature

# Shared branches off mainline/master/trunk are bad at any release cadence



# Trunk Based Development



# Branching strategy vs. tactics

Short-lived < 1 day

No *strategy*!

# Azure DevOps



# Azure DevOps



Azure  
Boards

Plan, track, and discuss work across teams, deliver value to your users faster.



Azure  
Repos

Unlimited cloud-hosted private Git repos. Collaborative pull requests, advanced file management, and more.



Azure  
Pipelines

CI/CD that works with any language, platform, and cloud. Connect to GitHub or any Git provider and deploy continuously to any cloud.



Azure  
Test Plans

The test management and exploratory testing toolkit that lets you ship with confidence.



Azure  
Artifacts

Create, host, and share packages. Easily add artifacts to CI/CD pipelines.



# Azure DevOps

Demo

# Continuous Integration Trunk Based Development

Demo

# Continuous Delivery

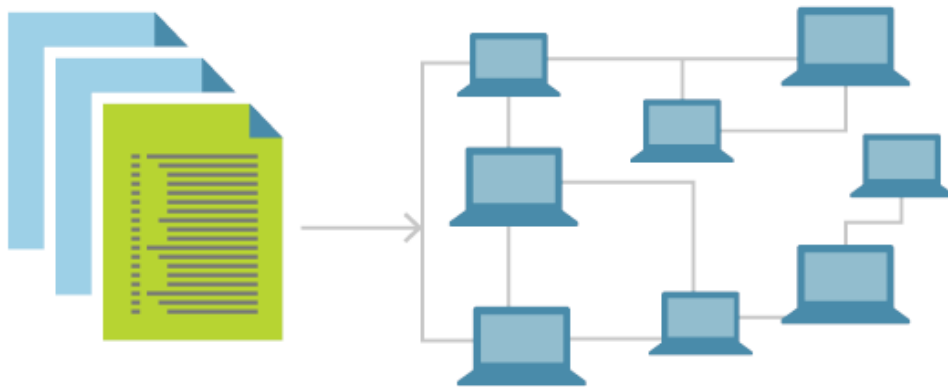
# Continuous Delivery

Continuous delivery is an approach where teams release quality products frequently and predictably from source code repository to production in an automated fashion.

# Continuous Deployment

# Infrastructure as Code

# Infrastructure as Code



Infrastructure as Code (IaC) is the management of infrastructure (networks, virtual machines, load balancers, and connection topology) in a descriptive model, using the same versioning as DevOps team uses for source code.

Idempotent

IaC is a key DevOps practice and is used in conjunction with CD

# Infrastructure as Code Tools for Microsoft Azure

Azure PowerShell

Azure CLI

ARM Templates

Terraform





# Continuous Delivery and Deployment Infrastructure as Code

Demo