
Life runs on code


It's a great time
to be a .NET developer

Microsoft Microsoft Azure Visual Studio

～ .NET Conf 2019 ～

.NET の今と今後に思うこと (Tokyo Ver.)

マイクロソフト コーポレーション
グローバル ブラックベルト
Azure Cloud Native テクニカル スペシャリスト

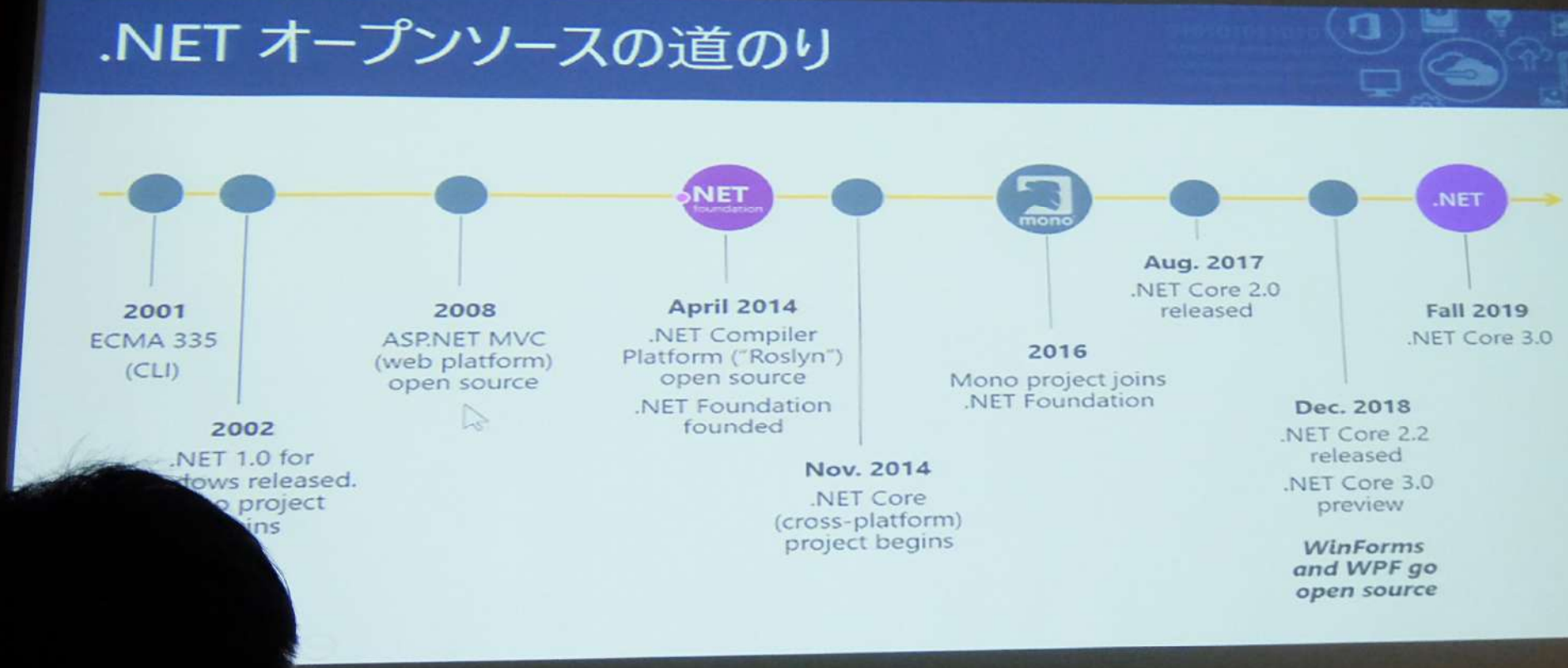
井上 章 (いのうえ あきら) @chack411


One ASP.NET ~ Katana Project
Application
One ASP.NET
Web Forms, MVC, Web API ...
ASP.NET
System.Web
IIS
Host

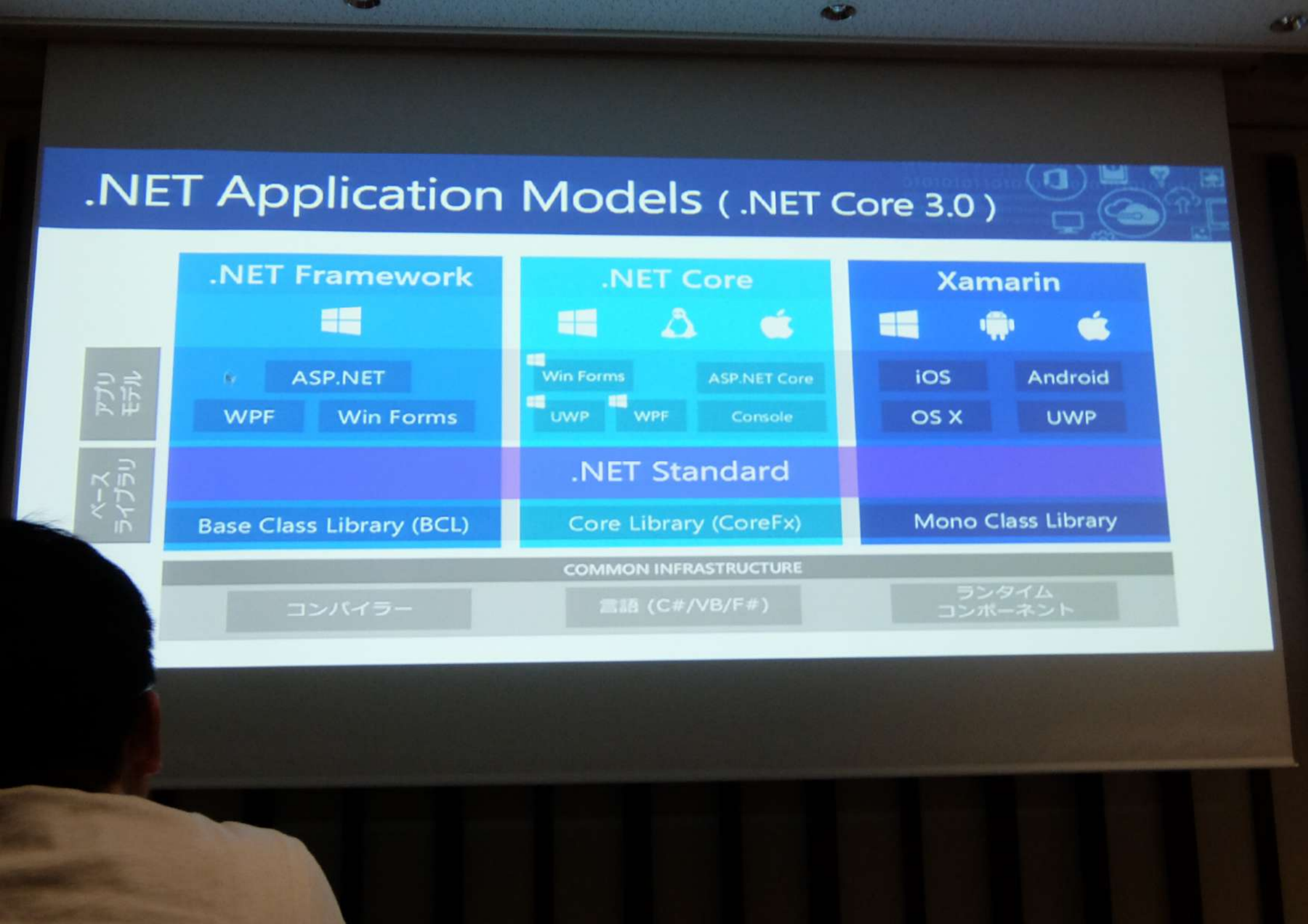
One ASP.NET ~ Katana Project

in 2014

.NET オープンソースの道のり




.NET foundation
CORPORATE SPONSORS
These companies are helping to drive the future of .NET
Google
Insight
JET BRAINS
Microsoft
Pivotal
redhat
SAMSUNG
Progress
Telerik
aws
unity


.NET Application Models (.NET Core 3.0)
.NET Framework
.NET Core
Xamarin
アプリモデル
ベースライブラリ
ASP.NET
WPF
Win Forms
Win Forms
ASP.NET Core
UWP
WPF
Console
iOS
Android
OS X
UWP
.NET Standard
Base Class Library (BCL)
Core Library (CoreFx)
Mono Class Library
COMMON INFRASTRUCTURE
コンパイラー
言語 (C#/VB/F#)
ランタイム
コンポーネント

.NET Framework の今後について

- **.NET Framework 4.8** が最後のメジャー バージョンとなる予定
- サポート ライフサイクル ポリシーは変更なし
  - インストール先の Windows OS と同じライフサイクル ポリシーが適用
  - Windows OS のコンポーネントとしてサポート (更新プログラム等)
  - (参考) ライフサイクルに関する FAQ – .NET Framework : https://support.microsoft.com/ja-jp/help/17455/lifecycle-faq-net-framework

既存 .NET Framework ベースのアプリケーションはそのまま利用可能
新規開発の場合は .NET Core を推奨

RELEASED

Visual Studio App Center for .NET Core 3.0 Windows Apps

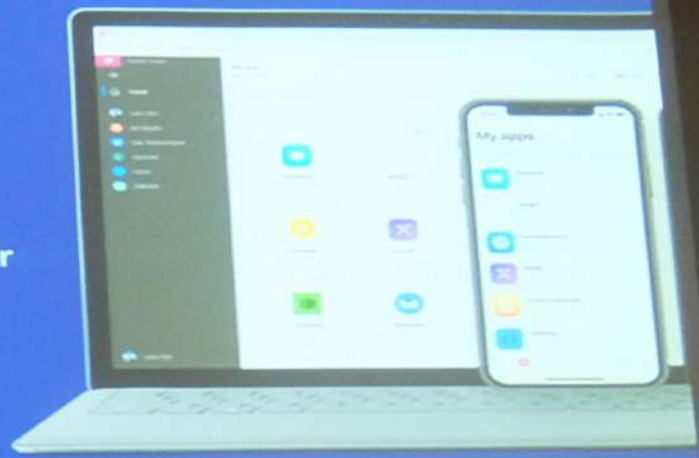
Easily deploy your app to your QA teams, testers, and app stores

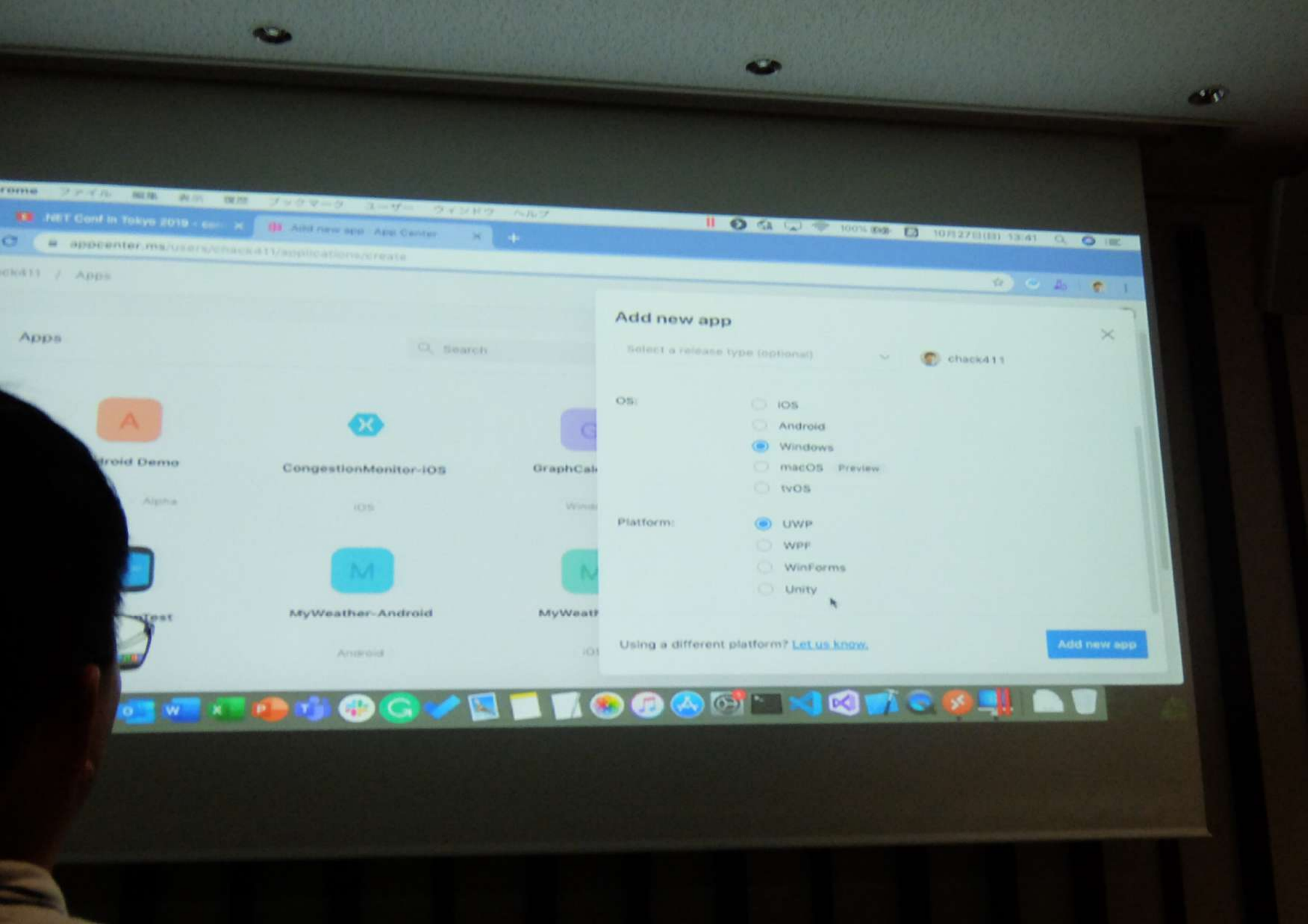
Collect real-time app diagnostics data, prioritize and fix issues faster

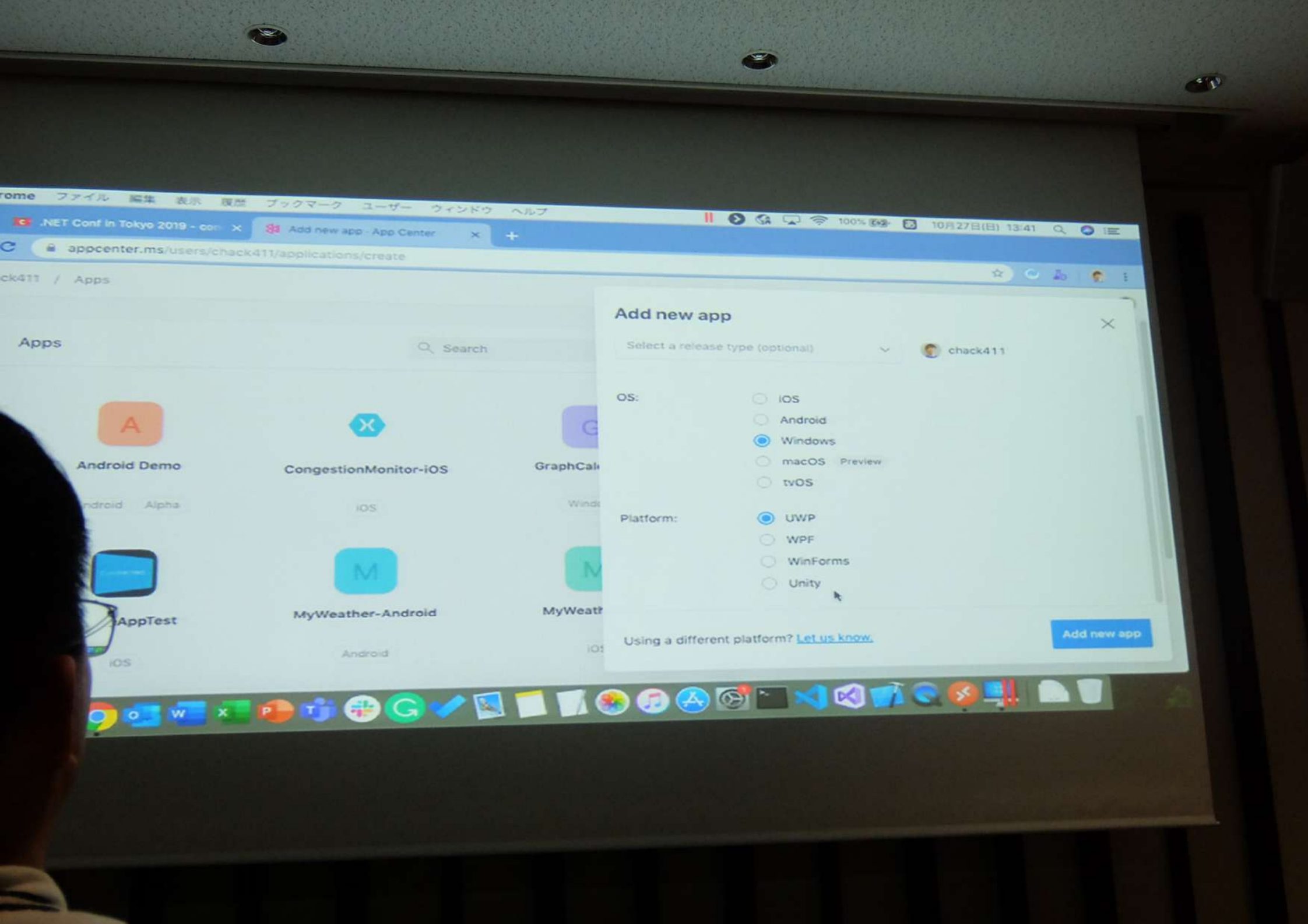
Trak usage patterns, user adoption, and other engagement metrics

Now supports WPF and WinForms apps on .NET Core 3.0

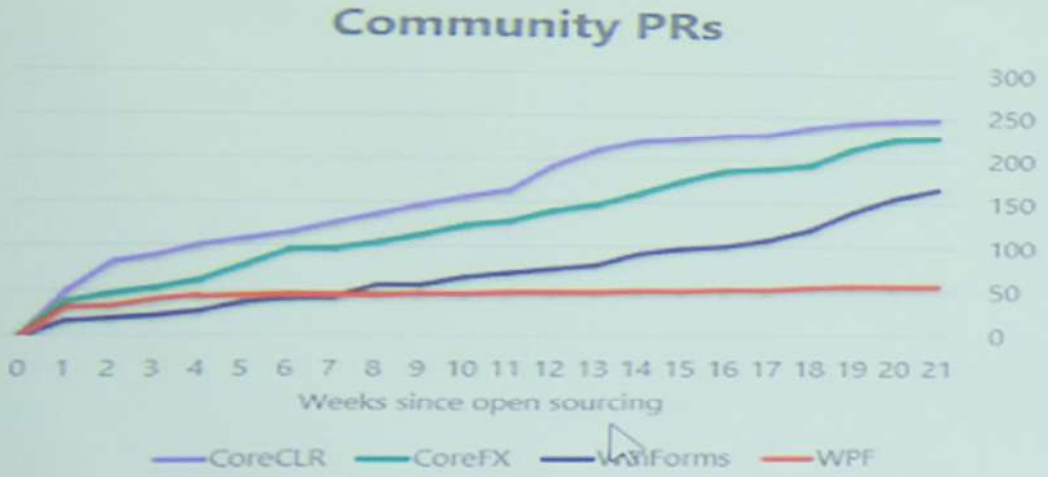
ppcenter.ms

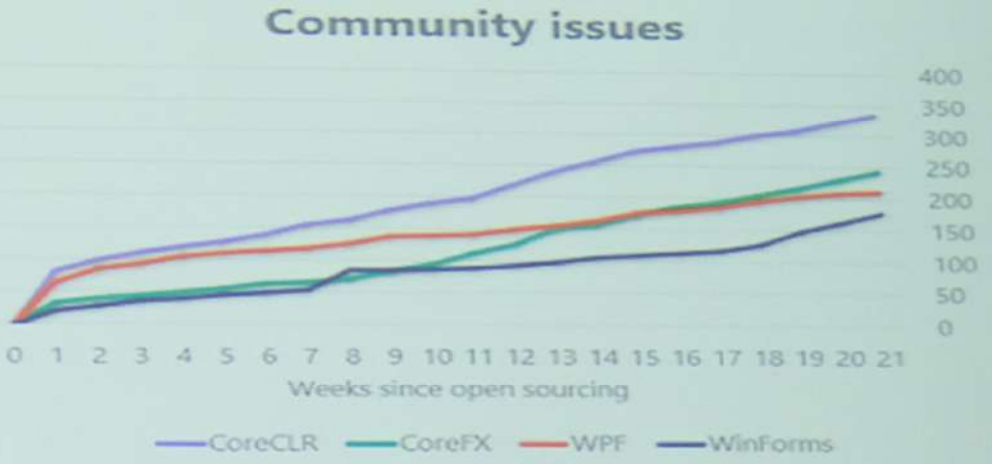



.NET Conf in Tokyo 2019
Add new app - App Center
appcenter.ms/users/chack411/applications/create
chack411 / Apps
Apps
Search
CongestionMeniter-iOS
iOS
MyWeather-Android
Android
Add new app
Select a release type (optional)
chack411
OS:
iOS
Android
Windows
macOS
Preview
tvOS
Platform:
UWP
WPF
WinForms
Unity
Using a different platform? Let us know.
Add new app


.NET Conf in Tokyo 2019 - con
Add new app - App Center
appcenter.ms/users/chack411/applications/create
ck411 / Apps
Apps
Search
Android Demo
CongestionMonitor-iOS
GraphCal
AppTest
MyWeather-Android
MyWeath
Add new app
Select a release type (optional)
chack411
OS:
iOS
Android
Windows
macOS Preview
tvOS
Platform:
UWP
WPF
WinForms
Unity
Using a different platform? Let us know.
Add new app

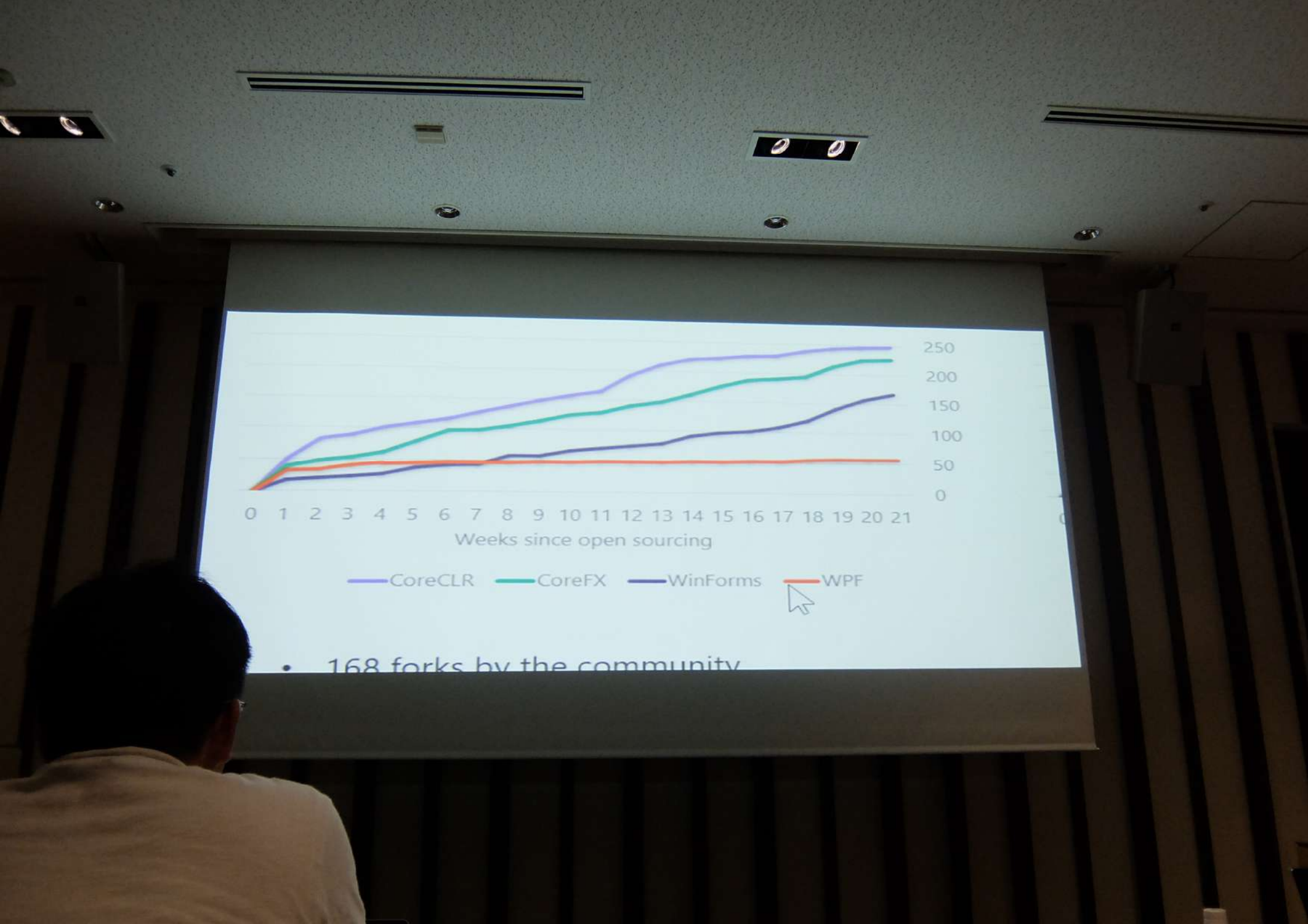
Windows Forms と WPF のオープンソース モメンタム





- 168 forks by the community
- 186 issues opened by the community
- 346 pull requests opened by the community

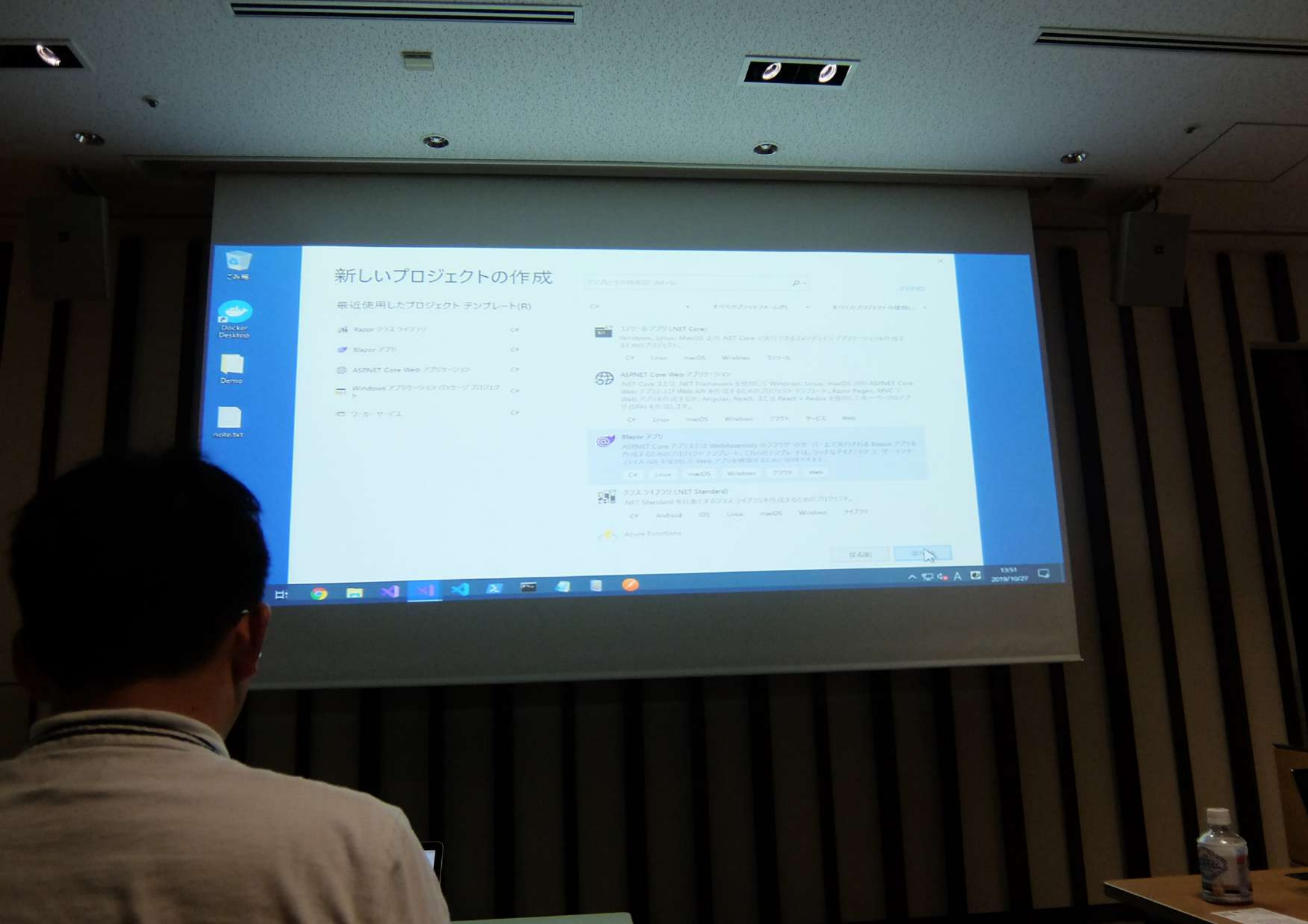
.NET foundation https://github.com/dotnet

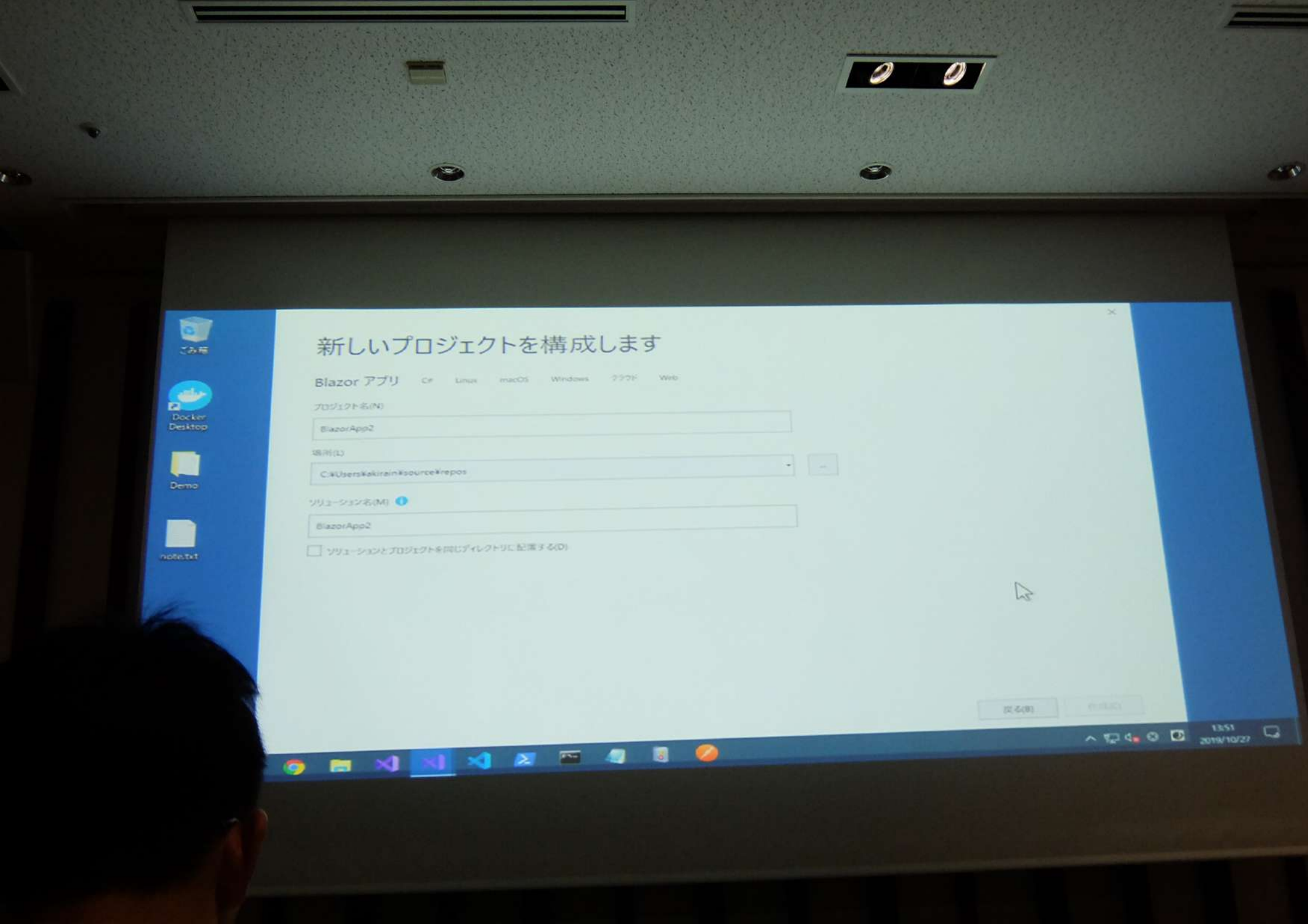

250
200
150
100
50
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
Weeks since open sourcing
CoreCLR
CoreFX
WinForms
WPF
• 168 forks by the community

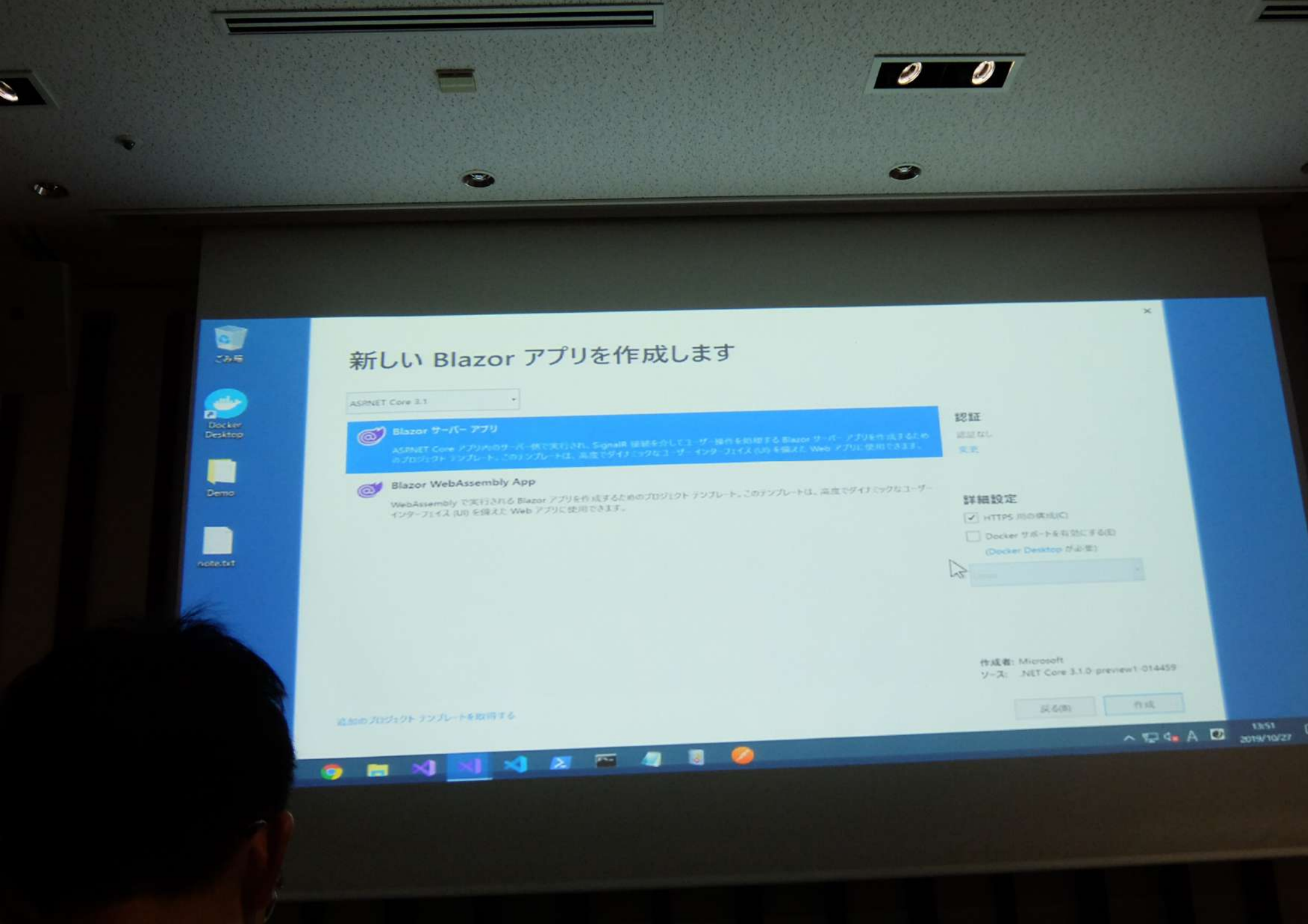
ASP.NET Core 3.0 Blazor

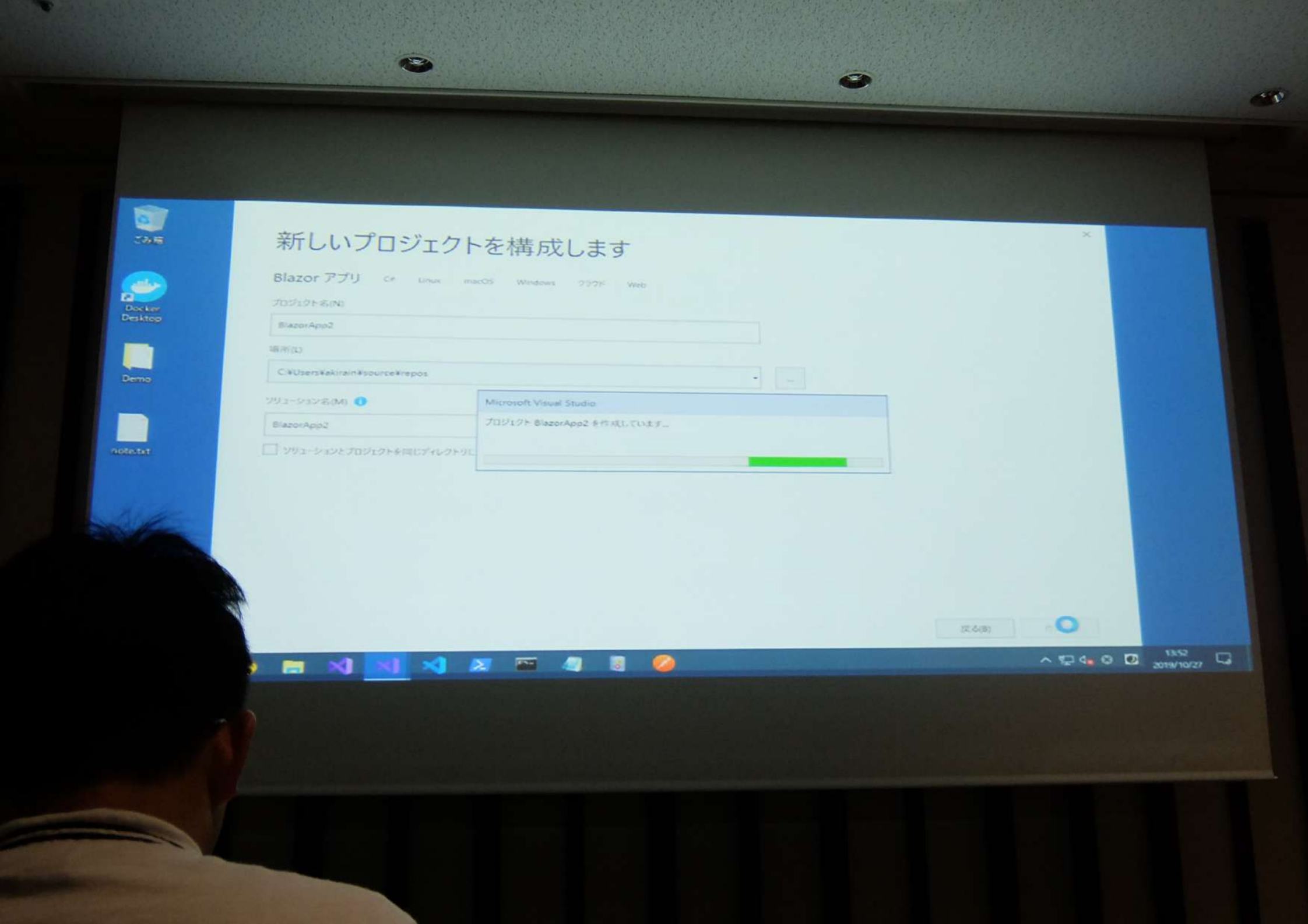
- **.NET (Razor Pages & C#) でフロントエンド Web UI を開発**
  - JavaScript, Angular, React, Vue などを知らなくても OK
  - .NET の安定性と一貫性
- **すべての WebAssembly 対応ブラウザーで動作**
  - ネイティブ パフォーマンス, プラグインなどは不要
  - Client と Server 間での C# コードの共有、強く型付けされた開発
- **2 種類のホスティング モデル**
  - **Blazor サーバー アプリ：**サーバーサイドの .NET Core プロセスで実行、SignalR で通信
    - .NET Core 3.0 で正式リリース
  - **Blazor WebAssembly アプリ：**Web ブラウザーの WebAssembly で実行
    - 現在プレビュー リリース (2020 年 5 月に正式リリース予定)

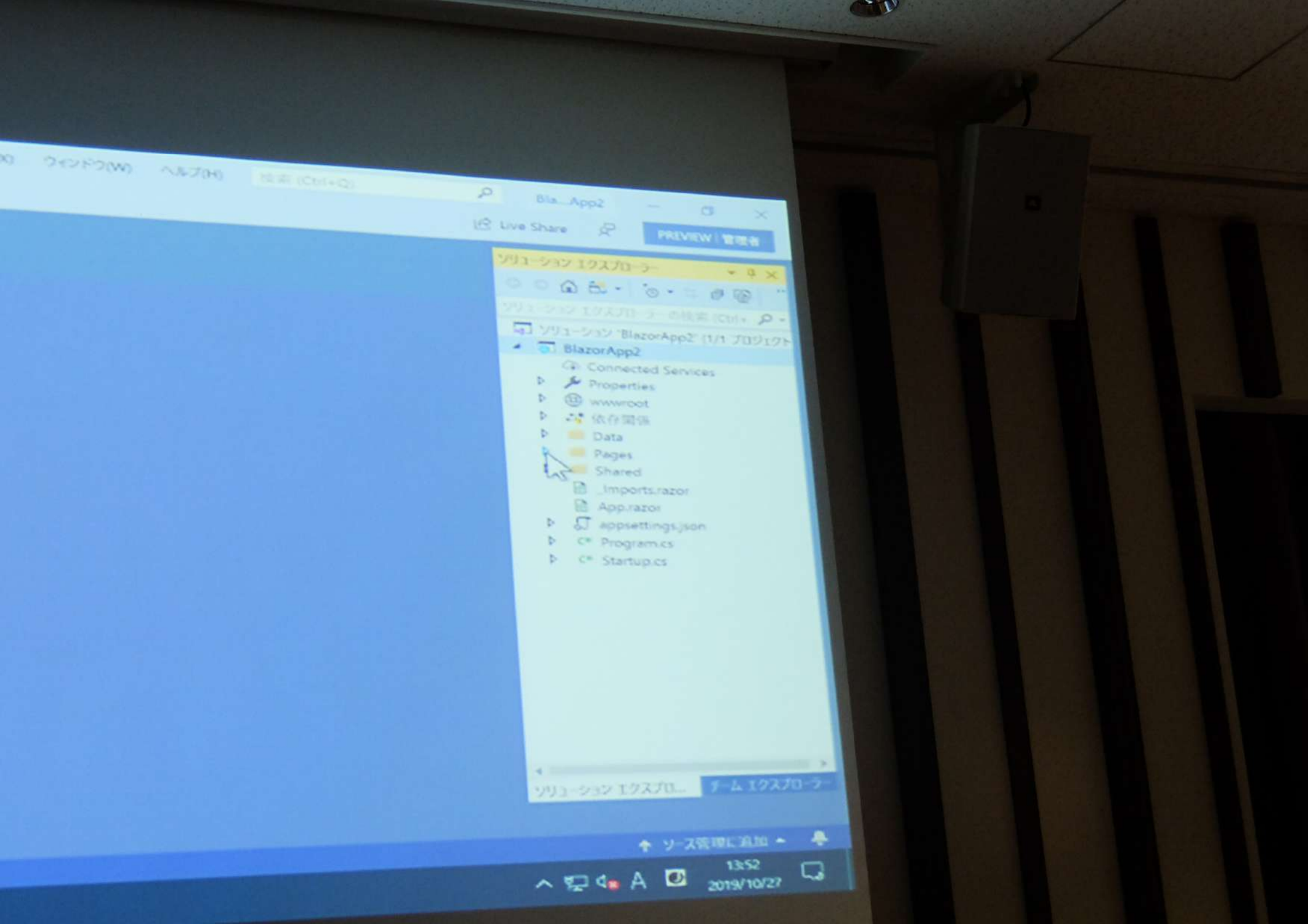
https://blazor.net

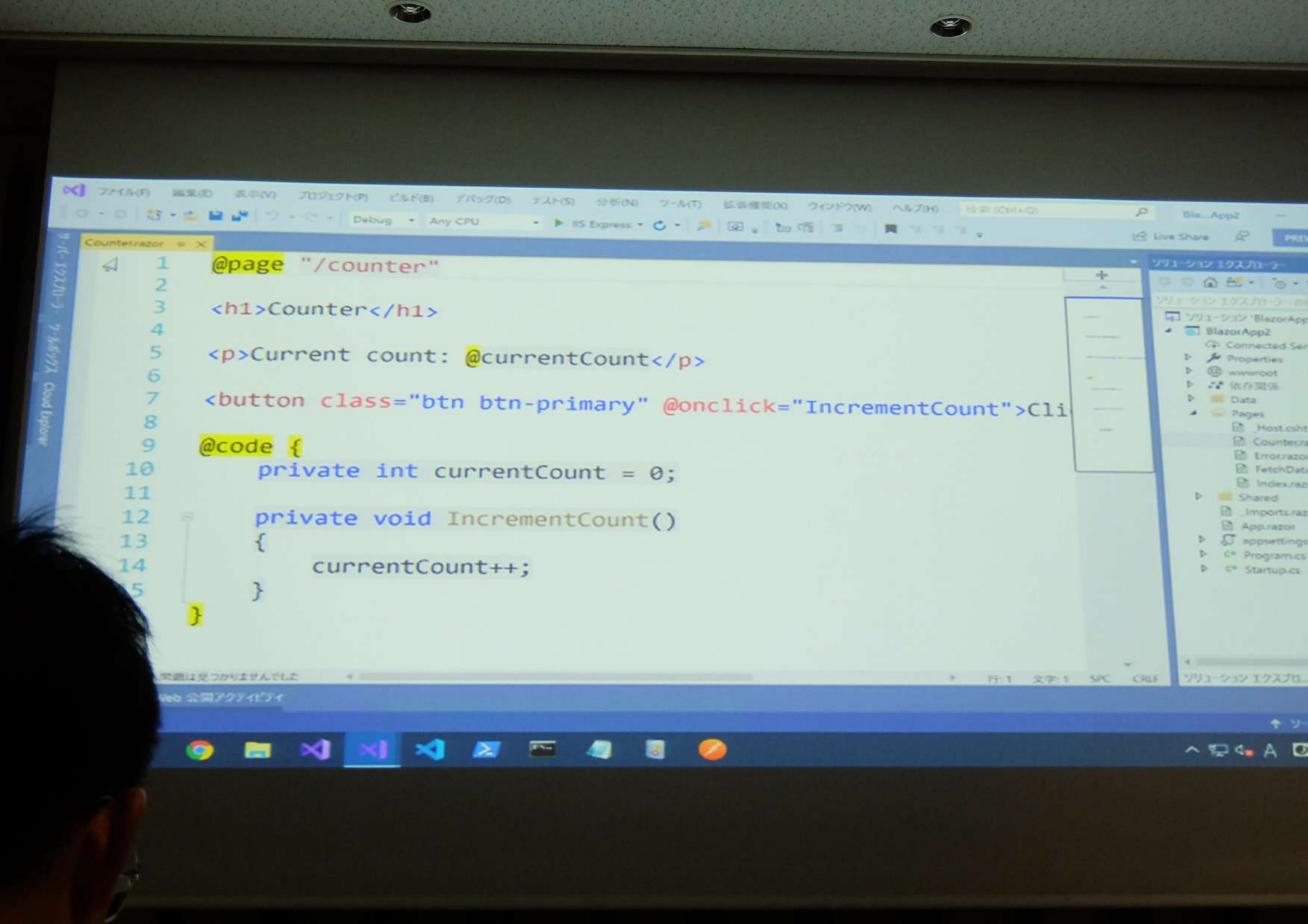

新しいプロジェクトの作成
最近使用したプロジェクト テンプレート(R)
ごみ箱
Docker Desktop
Demo
note.txt


新しいプロジェクトを構成します
Blazor アプリ
C#
Linux
macOS
Windows
クラウド
Web
プロジェクト名(N)
BlazorApp2
場所(L)
C:¥Users¥akirain¥source¥repos
ソリューション名(M)
BlazorApp2
ソリューションとプロジェクトを同じディレクトリに配置する(D)
戻る(B)
ごみ箱
Docker Desktop
Demo
note.txt


新しい Blazor アプリを作成します
ASP.NET Core 3.1
Blazor サーバー アプリ
Blazor WebAssembly App
認証
詳細設定
HTTPS 用の構成(C)
Docker サポートを有効にする(E)
(Docker Desktop が必要)
作成者: Microsoft
ソース: .NET Core 3.1.0 preview1.014459
戻る(B)
作成
追加のプロジェクト テンプレートを取得する
ごみ箱
Docker Desktop
Demo
note.txt

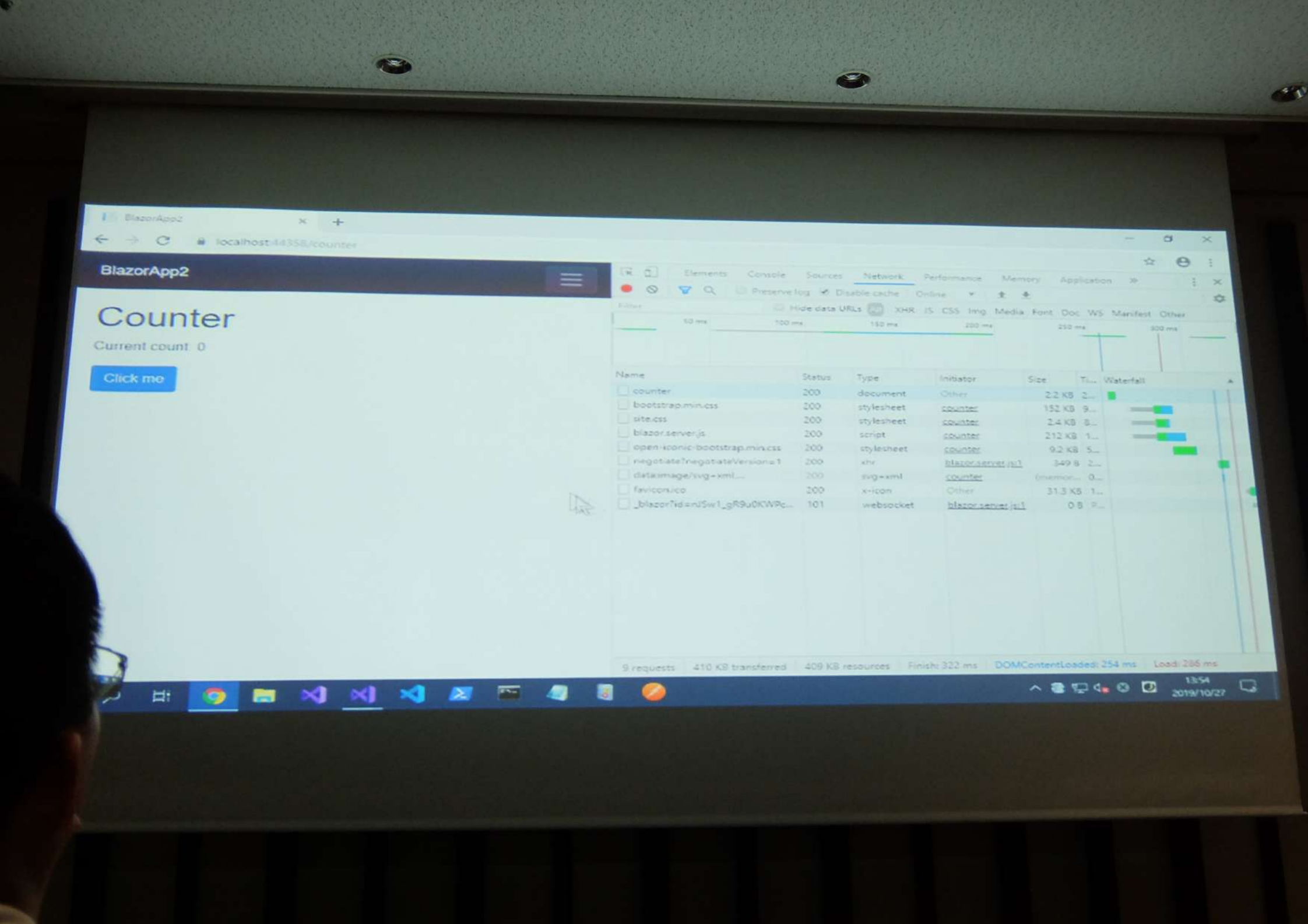

新しいプロジェクトを構成します
Blazor アプリ
C#
Linux
macOS
Windows
クラウド
Web
プロジェクト名(N)
BlazorApp2
場所(L)
C:¥Users¥akirain¥source¥repos
ソリューション名(M)
BlazorApp2
Microsoft Visual Studio
プロジェクト BlazorApp2 を作成しています...
戻る(B)
ごみ箱
Docker Desktop
Demo
note.txt

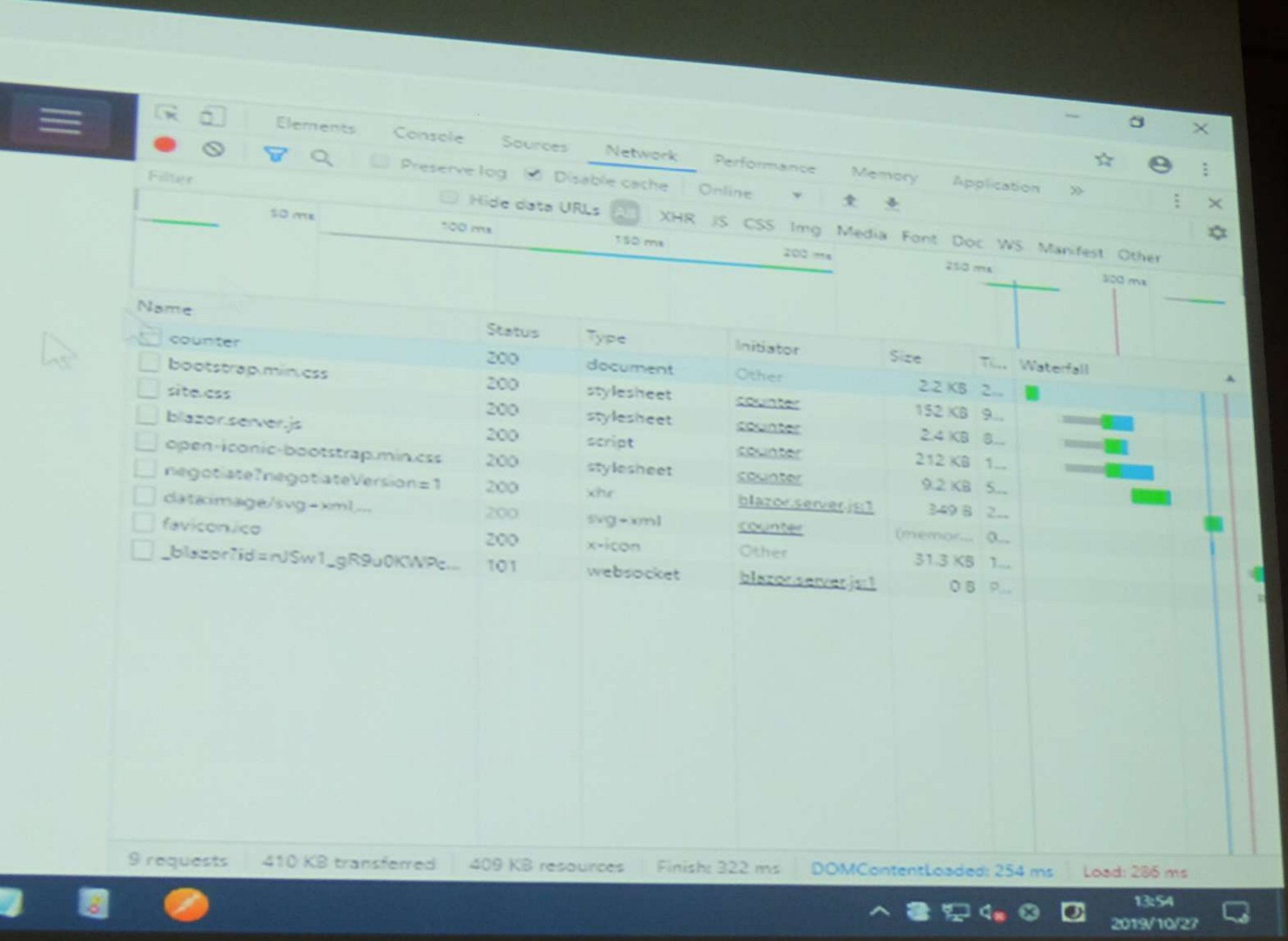

ウィンドウ(W)
ヘルプ(H)
Live Share
PREVIEW | 管理者
ソリューション エクスプローラー
ソリューション 'BlazorApp2' (1/1 プロジェクト
BlazorApp2
Connected Services
Properties
wwwroot
依存関係
Data
Pages
Shared
_Imports.razor
App.razor
appsettings.json
Program.cs
Startup.cs
ソリューション エクスプロ...
チーム エクスプローラー
ソース管理に追加
13:52
2019/10/27

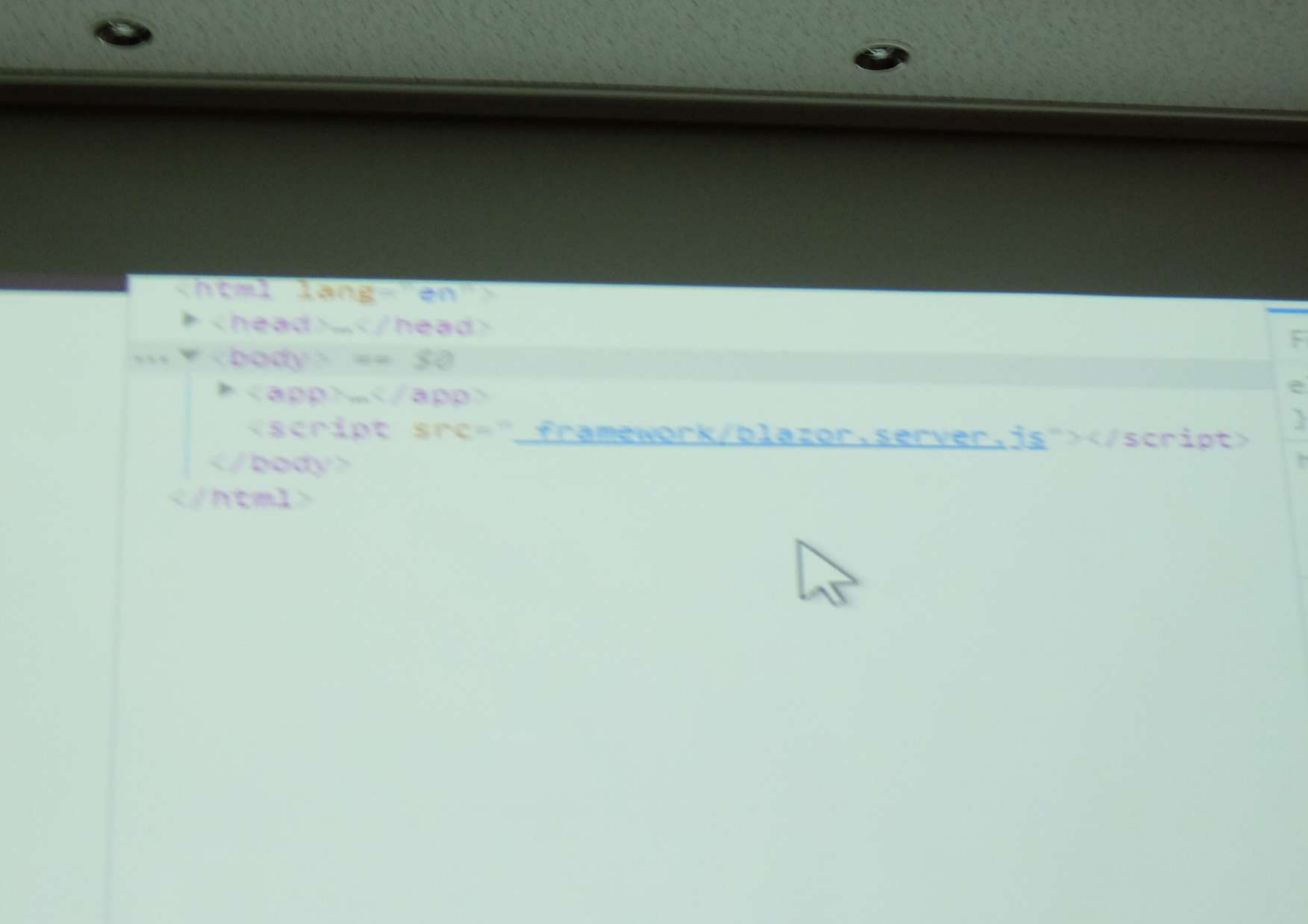

Counter.razor
@page "/counter"
<h1>Counter</h1>
Current count: @currentCount
<button class="btn btn-primary" @onclick="IncrementCount">Cli
@code {
private int currentCount = 0;
private void IncrementCount()
{
currentCount++;
}
}
Debug
Any CPU
IIS Express
Live Share
BlazorApp2
Properties
wwwroot
Data
Pages
Shared
App.razor
Program.cs
Startup.cs

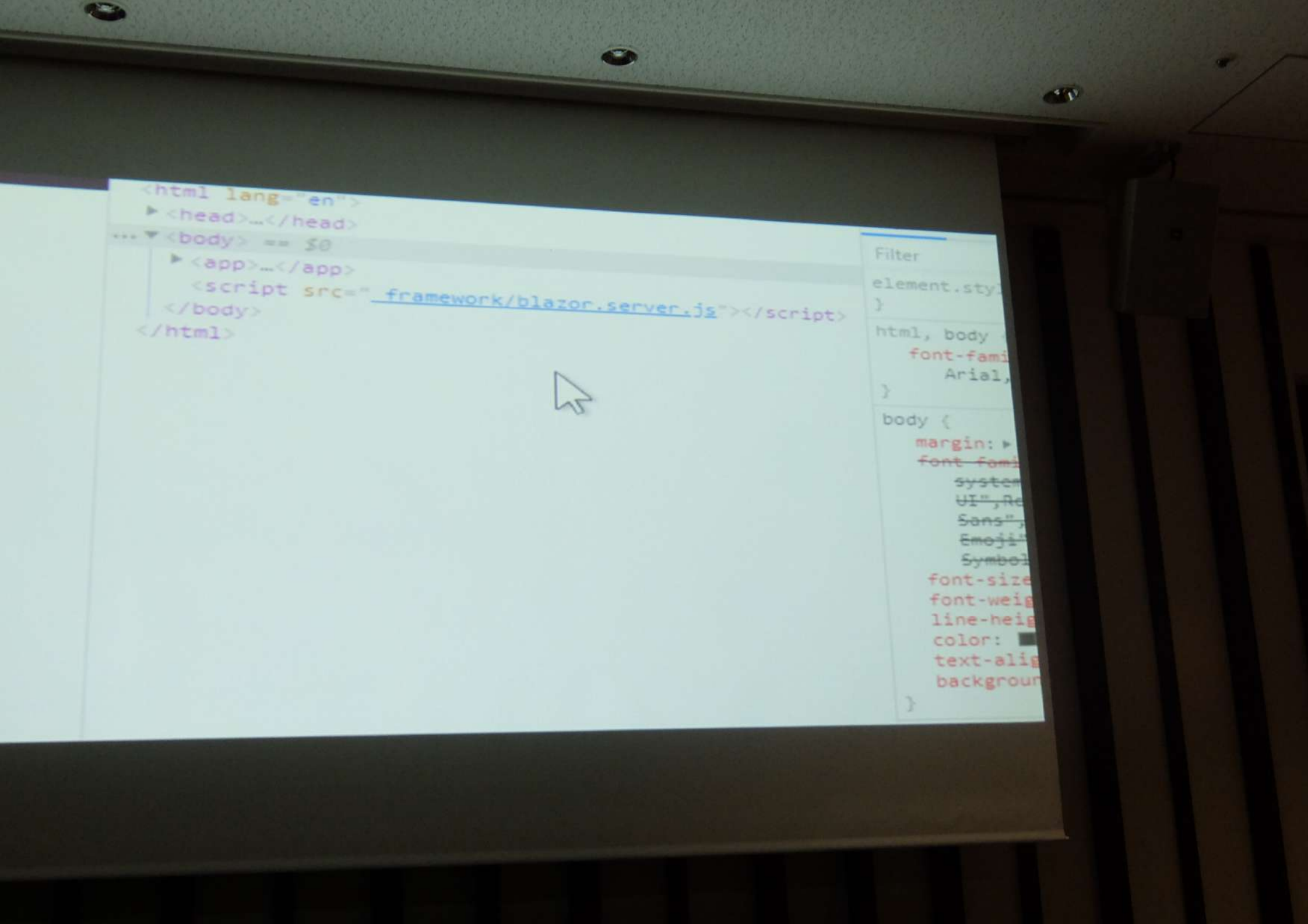
Counter.razor

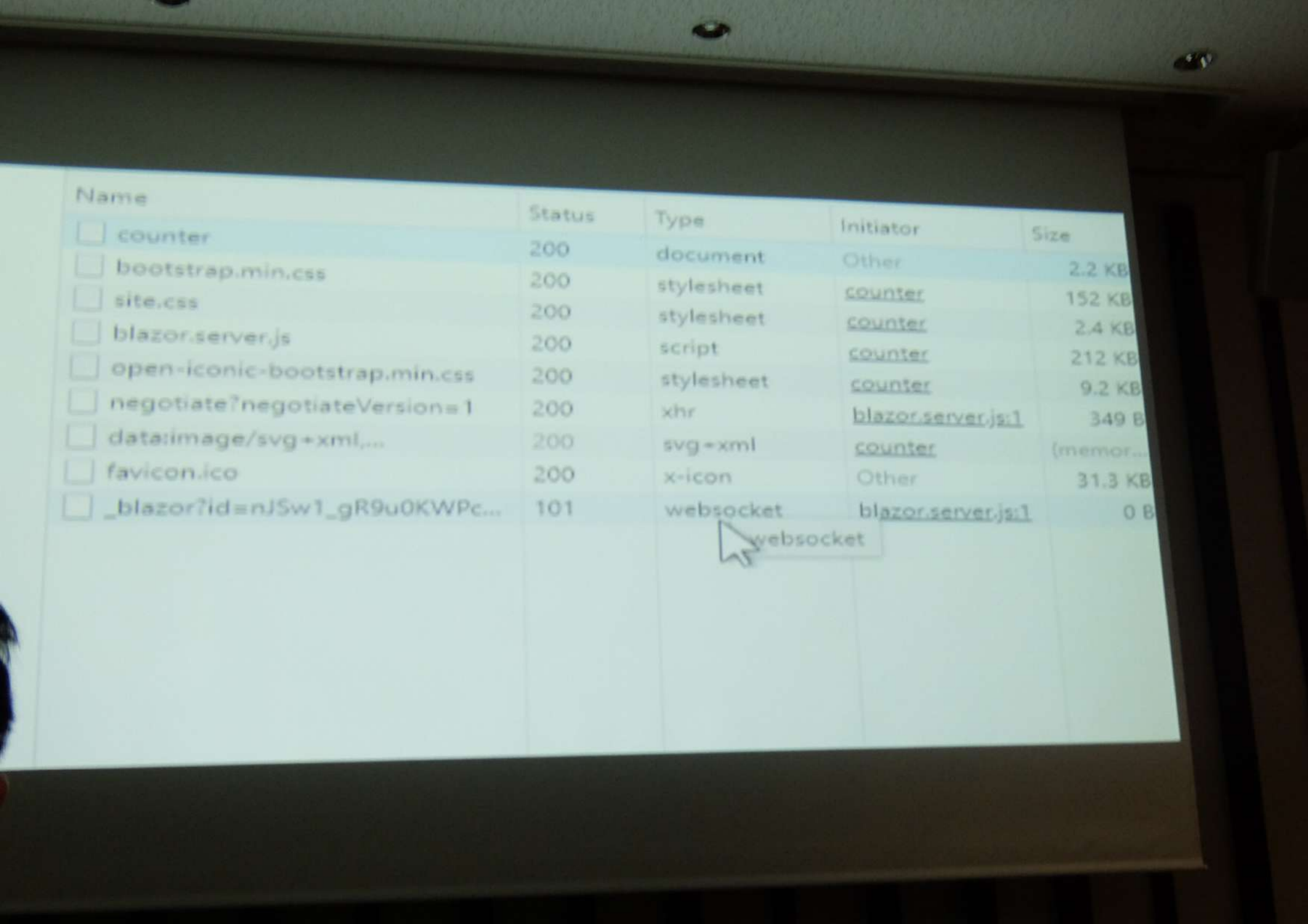
```
1   @page "/counter"
2
3   <h1>Counter</h1>
4
5   <p>Current count: @currentCount</p
6
7   <button class="btn btn-primary" @o
8
9   @code {
10      private int currentCount = 0;
11
```

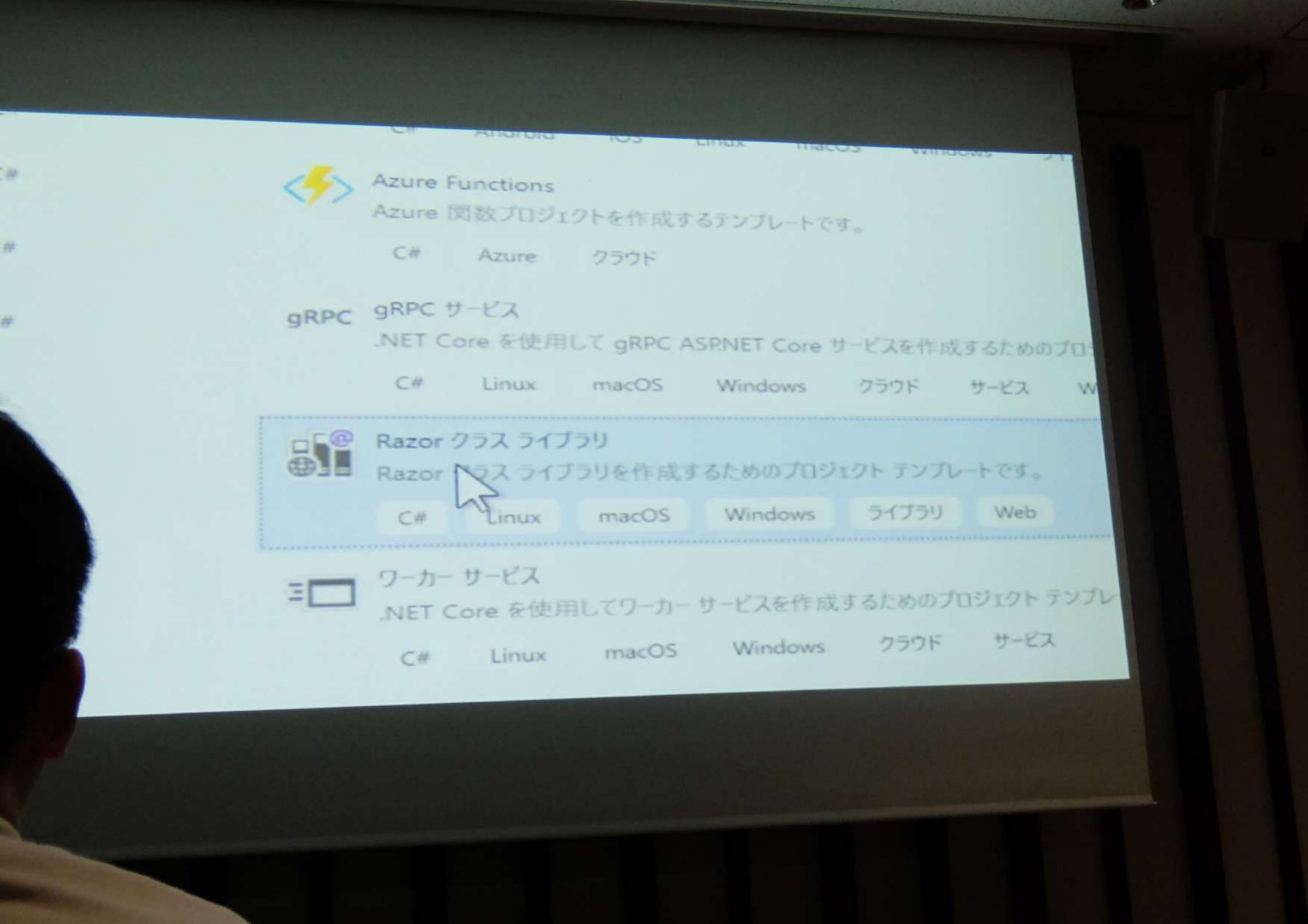

BlazorApp2
localhost:44358/counter
BlazorApp2
Counter
Current count: 0
Click me
Elements
Console
Sources
Network
Performance
Memory
Application
Name
Status
Type
Initiator
Size
Waterfall
counter
bootstrap.min.css
site.css
blazor.server.js
open-iconic-bootstrap.min.css
favicon.ico
9 requests
410 KB transferred
409 KB resources
Finish: 322 ms
DOMContentLoaded: 254 ms
Load: 286 ms

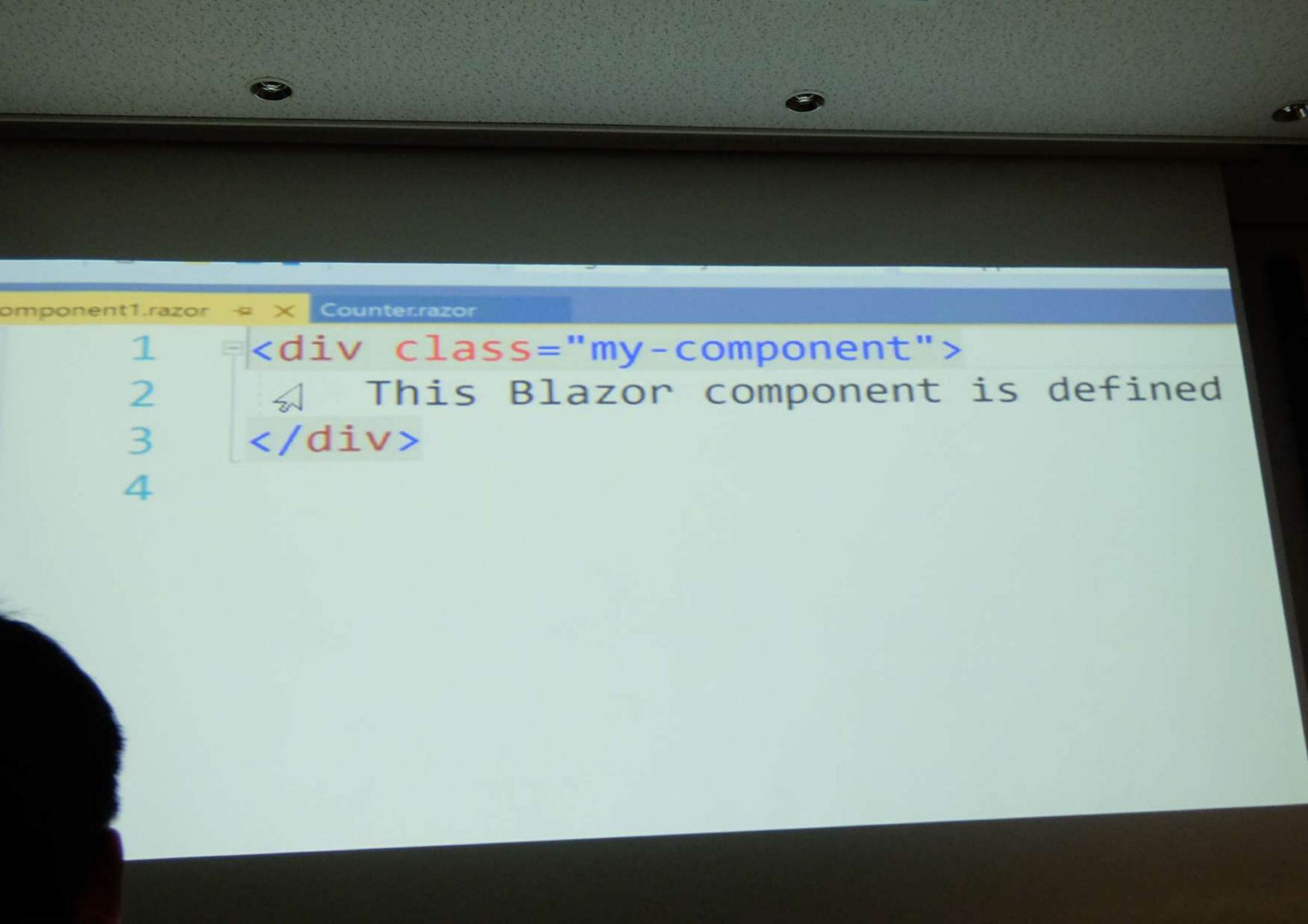

Elements
Console
Sources
Network
Performance
Memory
Application
Preserve log
Disable cache
Online
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
50 ms
100 ms
150 ms
200 ms
250 ms
300 ms
Name
Status
Type
Initiator
Size
Time
Waterfall
counter 200 document Other 2.2 KB 2...
bootstrap.min.css 200 stylesheet counter 152 KB 9...
site.css 200 stylesheet counter 2.4 KB 8...
blazor.server.js 200 script counter 212 KB 1...
open-iconic-bootstrap.min.css 200 stylesheet counter 9.2 KB 5...
negotiate?negotiateVersion=1 200 xhr blazor.server.js:1 349 B 2...
data:image/svg+xml,... 200 svg+xml counter (memor... 0...
favicon.ico 200 x-icon Other 31.3 KB 1...
_blazor?id=nJSw1_gR9u0KWPc... 101 websocket blazor.server.js:1 0 B P...
9 requests
410 KB transferred
409 KB resources
Finish: 322 ms
DOMContentLoaded: 254 ms
Load: 286 ms
13:54
2019/10/27

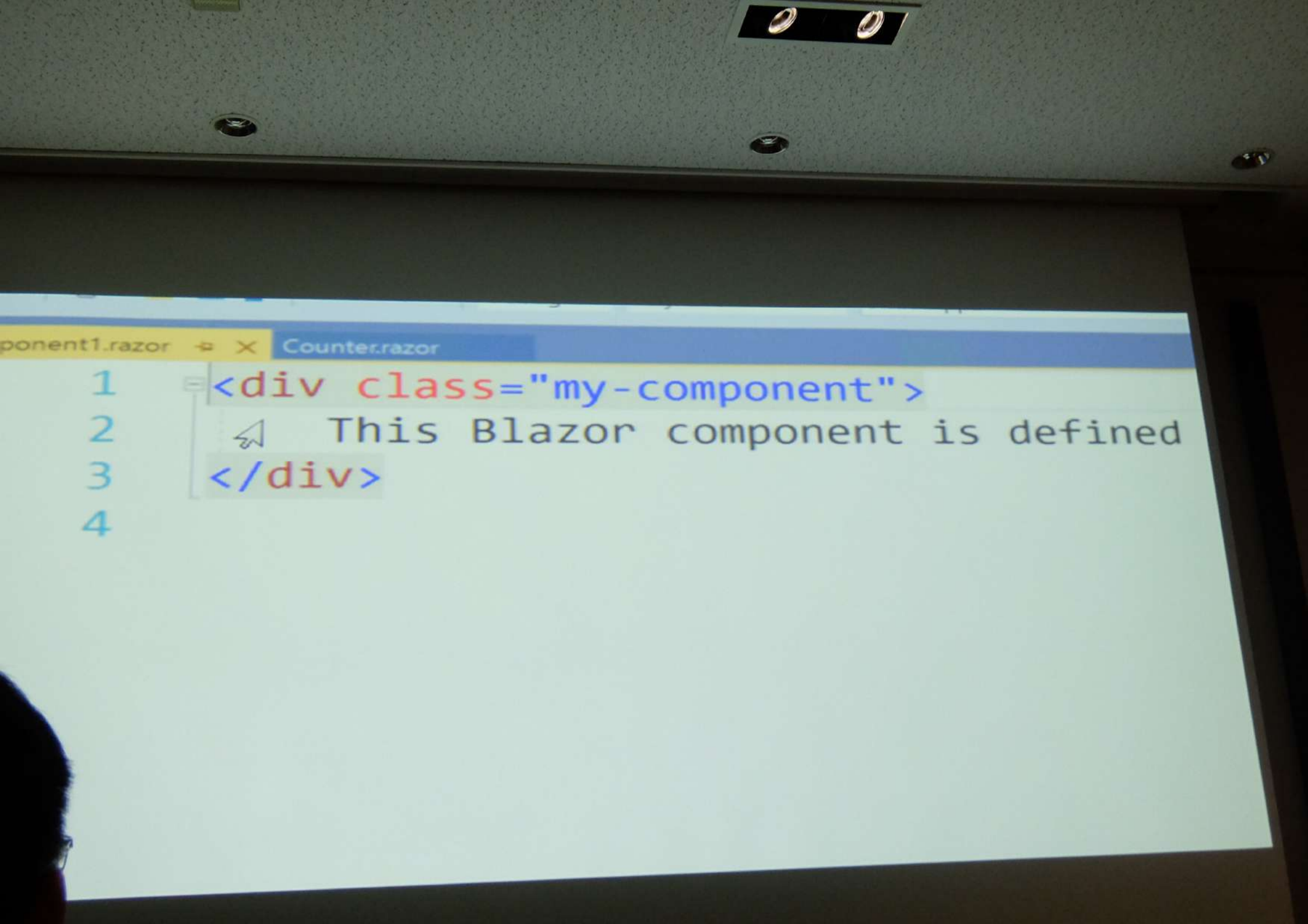

<html lang="en">
<head>…</head>
<body> == $0
<app>…</app>
<script src="_framework/blazor.server.js"></script>
</body>
</html>

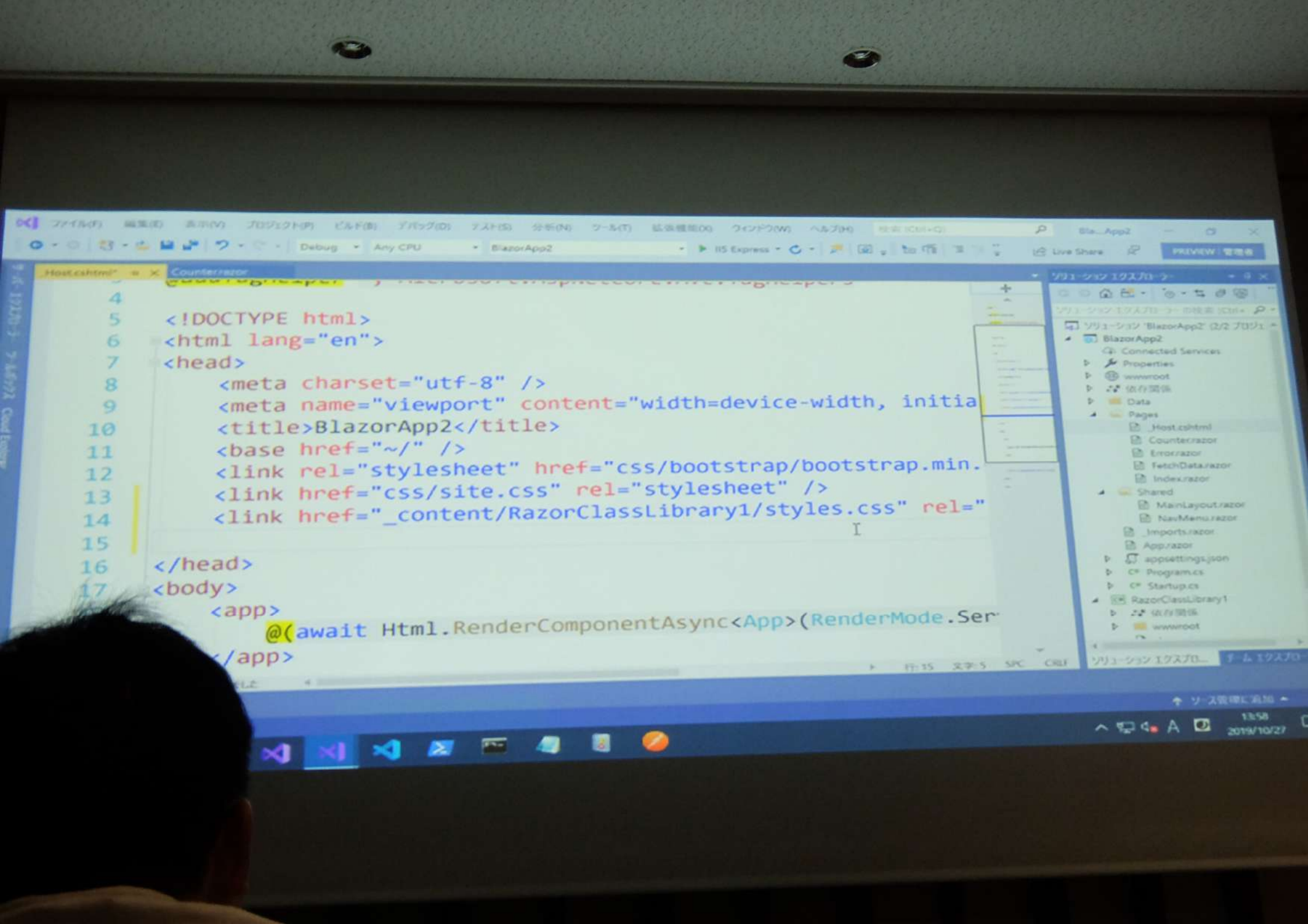

<html lang="en">
<head>…</head>
<body> == $0
<app>…</app>
<script src="_framework/blazor.server.js"></script>
</body>
</html>
Filter
element.sty
}
html, body
font-fami
Arial,
}
body {
margin:
font-fami
syste
UI",Ne
Sans",
Emoji"
Symbol
font-size
font-weig
line-heig
color:
text-alig
backgrou
}

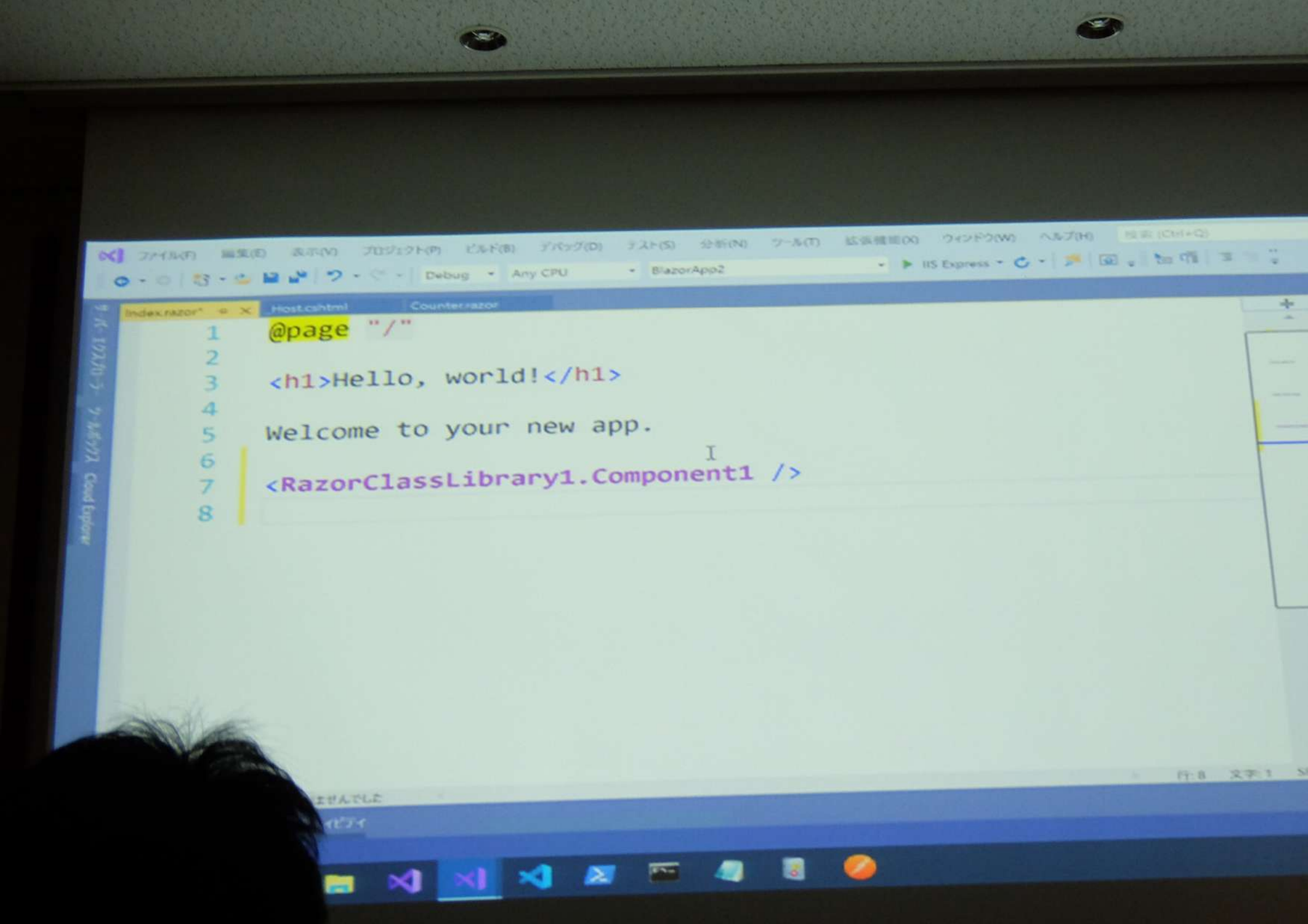

Name
Status
Type
Initiator
Size
counter
200
document
Other
2.2 KB
bootstrap.min.css
200
stylesheet
counter
152 KB
site.css
200
stylesheet
counter
2.4 KB
blazor.server.js
200
script
counter
212 KB
open-iconic-bootstrap.min.css
200
stylesheet
counter
9.2 KB
negotiate?negotiateVersion=1
200
xhr
blazor.server.js:1
349 B
data:image/svg+xml,...
200
svg+xml
counter
(memor...
favicon.ico
200
x-icon
Other
31.3 KB
_blazor?id=nJSw1_gR9u0KWPc...
101
websocket
blazor.server.js:1
0 B
websocket

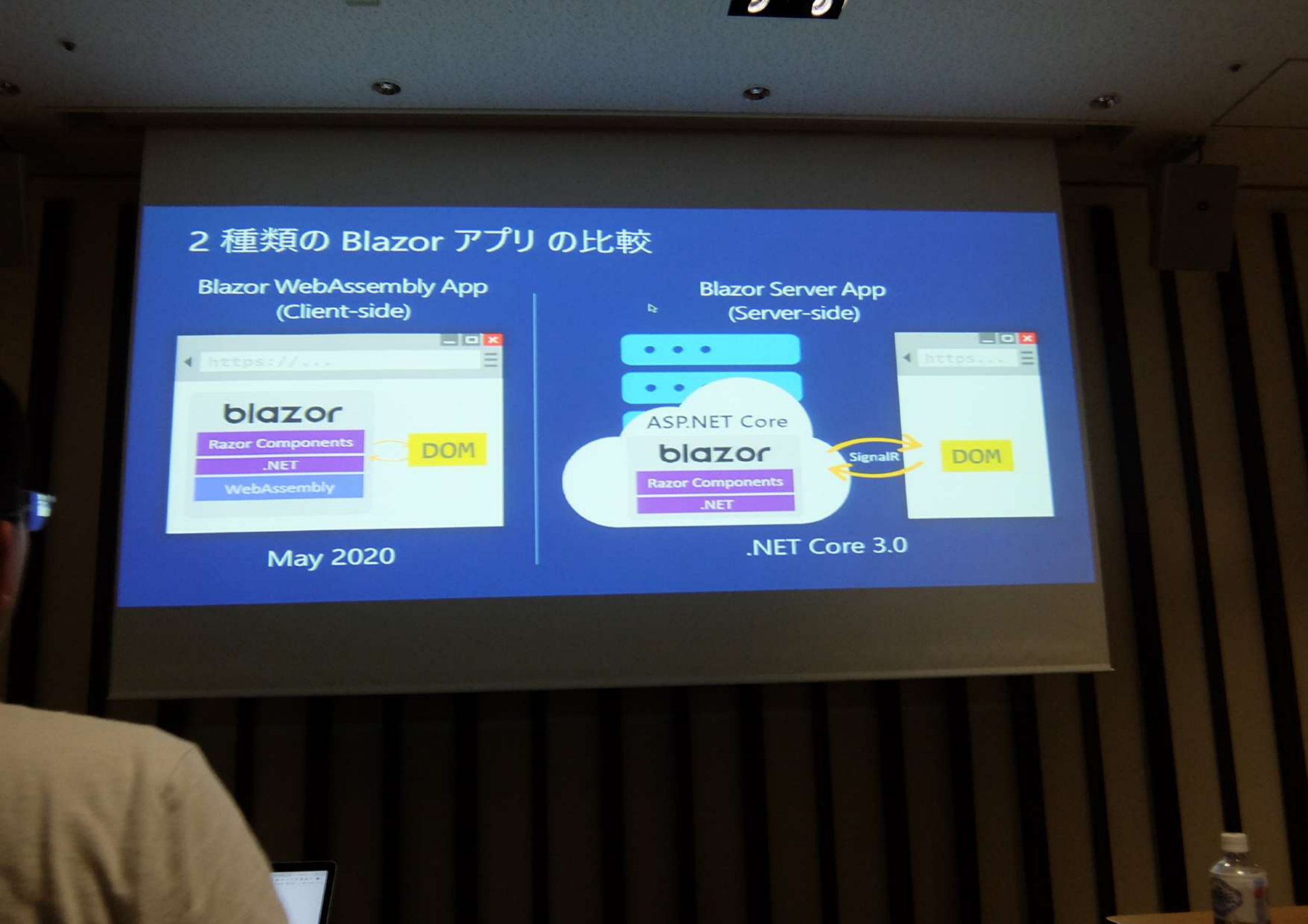

Azure Functions
Azure 関数プロジェクトを作成するテンプレートです。
C# Azure クラウド
gRPC
gRPC サービス
.NET Core を使用して gRPC ASP.NET Core サービスを作成するためのプロ
C# Linux macOS Windows クラウド サービス
Razor クラス ライブラリ
Razor クラス ライブラリを作成するためのプロジェクト テンプレートです。
C# Linux macOS Windows ライブラリ Web
ワーカー サービス
.NET Core を使用してワーカー サービスを作成するためのプロジェクト テンプレ
C# Linux macOS Windows クラウド サービス


omponent1.razor
Counter.razor
1
2 This Blazor component is defined
3
4


ponent1.razor
Counter.razor
1
2 This Blazor component is defined
3
4

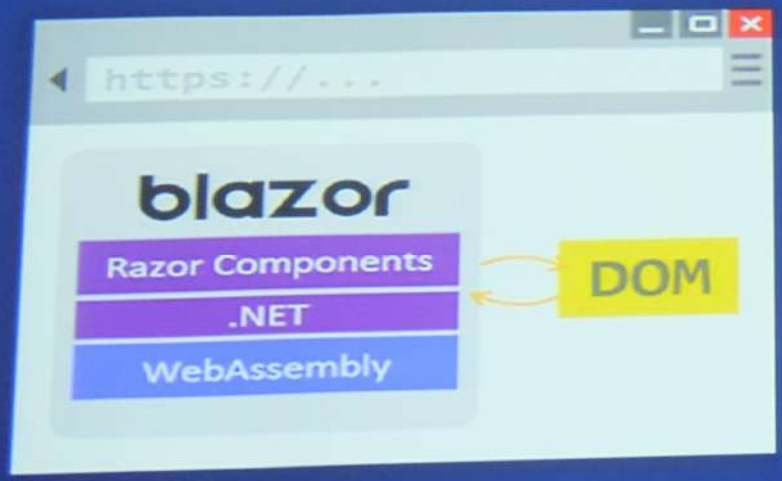

Host.cshtml
Counter.razor
Debug
Any CPU
BlazorApp2
IIS Express
Live Share
PREVIEW
4
5 <!DOCTYPE html>
6 <html lang="en">
7 <head>
8 <meta charset="utf-8" />
9 <meta name="viewport" content="width=device-width, initia
10 <title>BlazorApp2</title>
11 <base href="~/" />
12 <link rel="stylesheet" href="css/bootstrap/bootstrap.min.
13 <link href="css/site.css" rel="stylesheet" />
14 <link href="_content/RazorClassLibrary1/styles.css" rel="
15
16 </head>
17 <body>
<app>
@(await Html.RenderComponentAsync<App>(RenderMode.Ser
</app>
ソリューション エクスプローラー
ソリューション 'BlazorApp2' (2/2 プロジェ
BlazorApp2
Connected Services
Properties
wwwroot
Data
Pages
_Host.cshtml
Counter.razor
Error.razor
FetchData.razor
Index.razor
Shared
MainLayout.razor
NavMenu.razor
_Imports.razor
App.razor
appsettings.json
Program.cs
Startup.cs
RazorClassLibrary1
wwwroot
ソース管理に追加
13:58
2019/10/27


Debug
Any CPU
BlazorApp2
IIS Express
Index.razor*
Host.cshtml
Counter.razor
@page "/"
<h1>Hello, world!</h1>
Welcome to your new app.
<RazorClassLibrary1.Component1 />
8


2 種類の Blazor アプリ の比較
Blazor WebAssembly App
(Client-side)
https://...
blazor
Razor Components
.NET
WebAssembly
DOM
May 2020
Blazor Server App
(Server-side)
ASP.NET Core
blazor
Razor Components
.NET
SignalR
https...
DOM
.NET Core 3.0

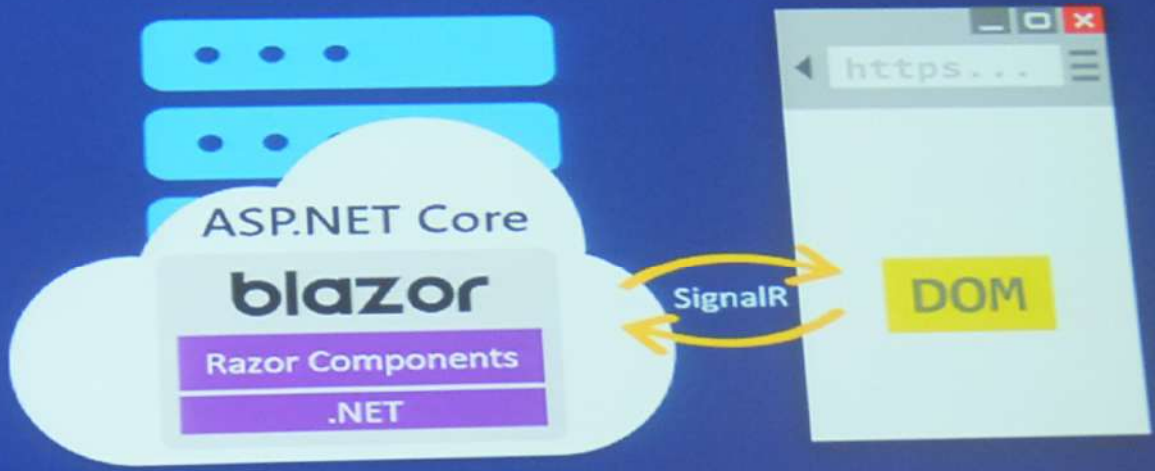
2 種類の Blazor アプリ の比較

Blazor WebAssembly App (Client-side)



May 2020

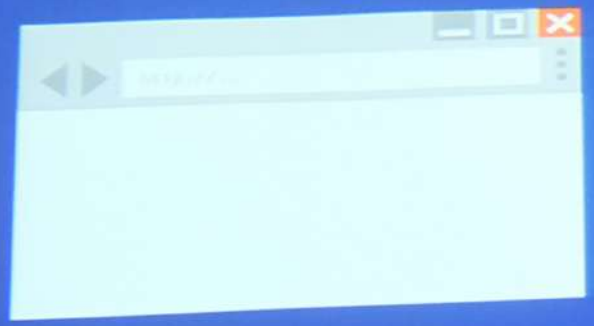
Blazor Server App (Server-side)



.NET Core 3.0

.NET Core 3.0 : フルスタック Web Development

Client



- Blazor

Frontend

ASP.NET Core

- MVC / Razor Pages
- Web APIs
- SignalR
- Security & identity

.NET Core 3.0 : フルスタック Web Development

Client

- Blazor
- Components
- SPA (JavaScript)

Frontend

- MVC / Razor Pages
- Web APIs
- SignalR
- Security & identity

.NET Core 3.0 : フルスタック Web Development

Client

- Blazor
- Components
- SPA (JavaScript)

Frontend

- MVC / Razor Pages
- Web APIs
- SignalR
- Security & identity

Backend

- Worker services
- gRPC

.NET Core 3.0 : フルスタック Web Development

Client

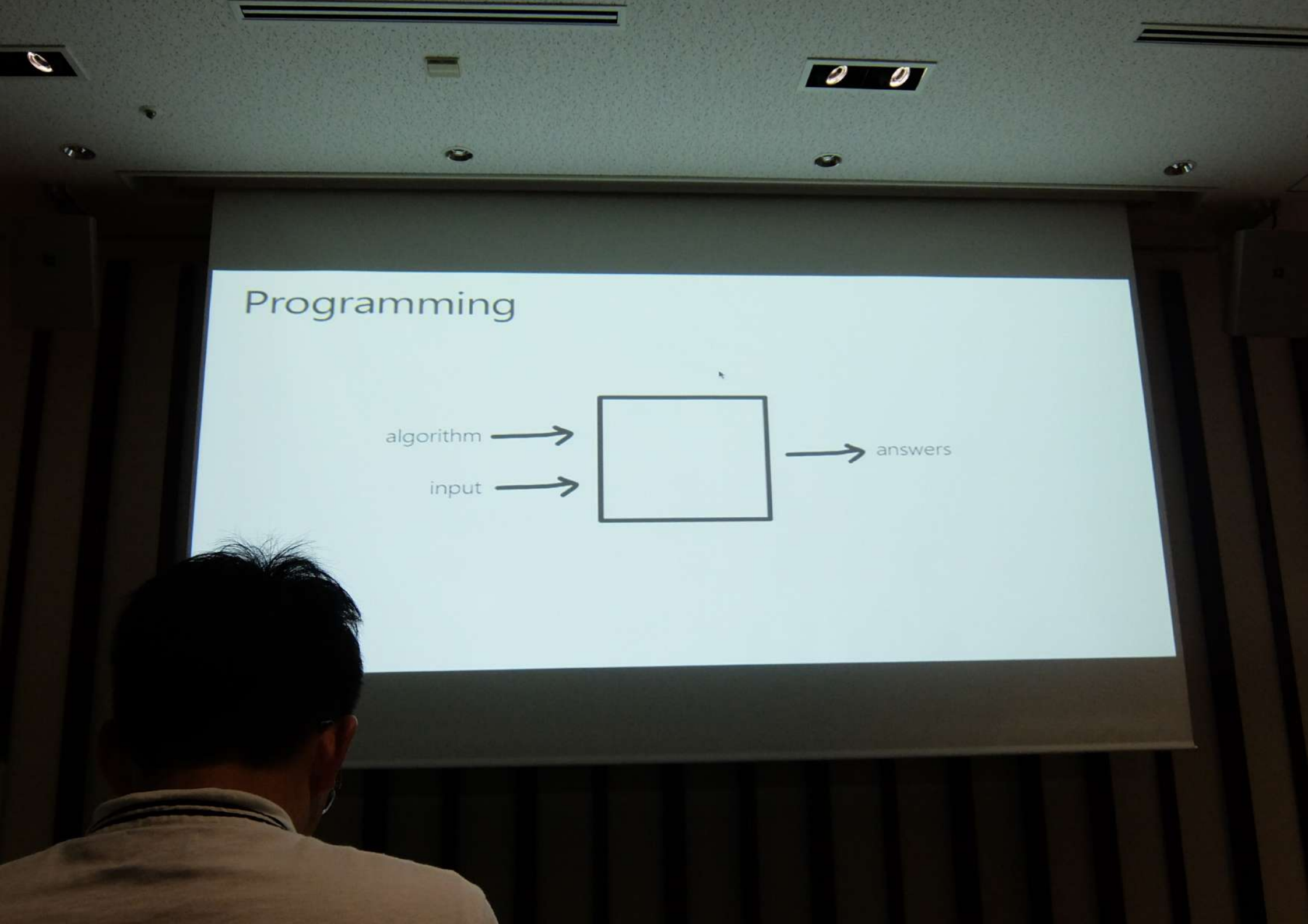
- Blazor
- Components
- SPA (JavaScript)

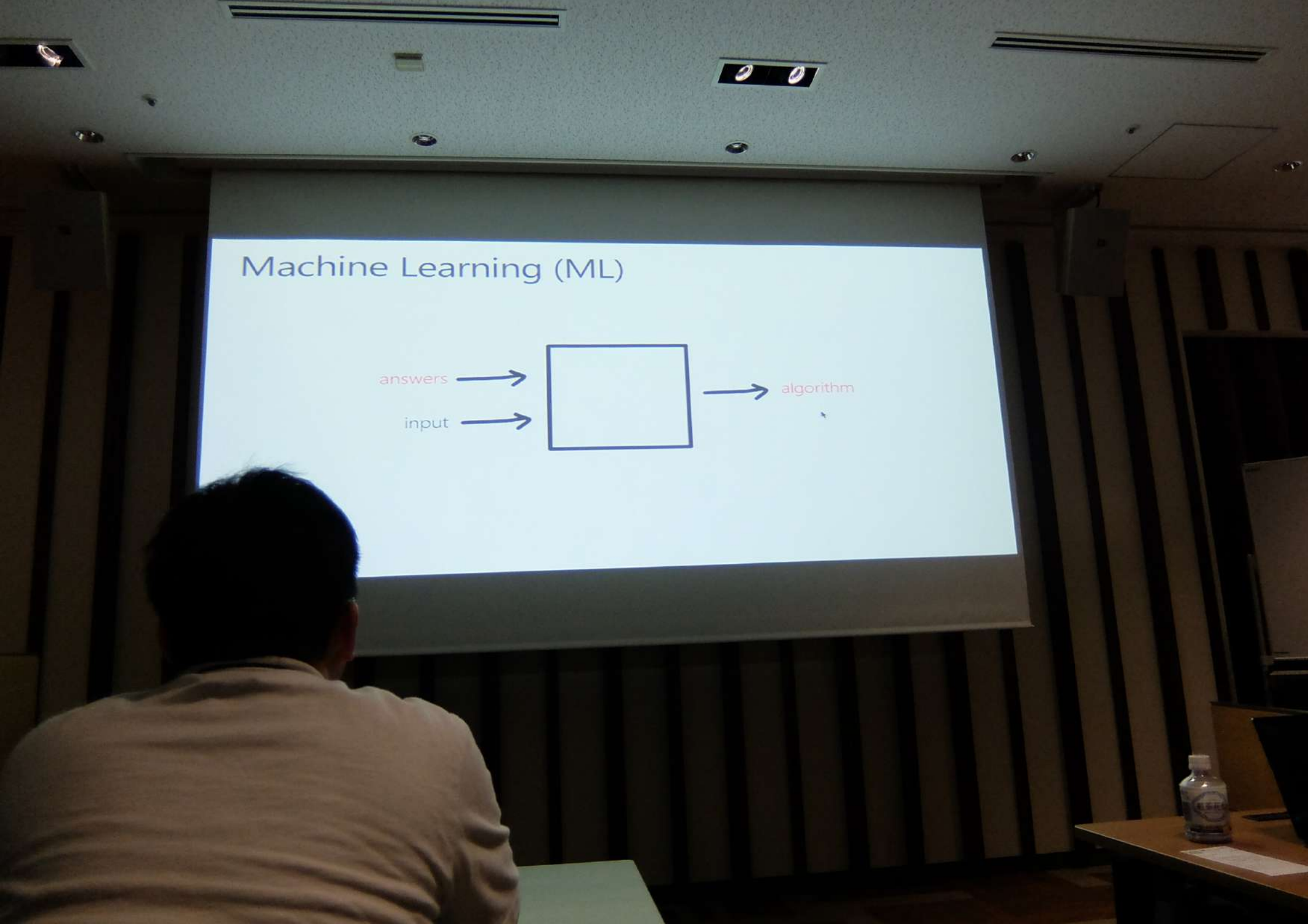
Frontend

- MVC / Razor Pages
- Web APIs
- SignalR
- Security & identity

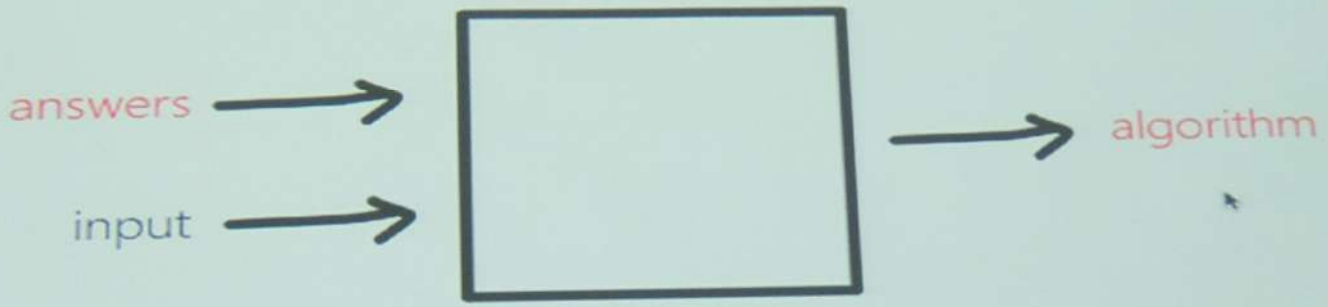
Backend

- Worker services
- gRPC


Programming
algorithm
input
answers


Machine Learning (ML)
answers
input
algorithm

Machine Learning (ML)



ML.NET 使用例

Sentiment analysis
Analyze the sentiment of customer reviews using a binary classification algorithm.

Product recommendation
Recommend products based on purchase history using a matrix factorization algorithm.

Price prediction
Predict taxi fares based on distance traveled etc. using a regression algorithm.

Customer segmentation
Identify groups of customers with similar profiles using a clustering algorithm.

GitHub labeler
Suggest the GitHub label for new issues using a multi-class classification algorithm.

Fraud detection
Detect fraudulent credit card transactions using a binary classification algorithm.

Spam detection
Flag text messages as spam using a binary classification algorithm.

Image classification
Classify images (e.g. broccoli vs pizza) using a TensorFlow deep learning algorithm.

Sales forecasting
Forecast future sales for products using a regression algorithm.

dot.net/ml

github.com/dotnet/machinelearning-samples

ML.NET 使用例

Sentiment analysis
Analyze the sentiment of customer reviews using a binary classification algorithm.

Product recommendation
Recommend products based on purchase history using a matrix factorization algorithm.

Price prediction
Predict taxi fares based on distance traveled etc. using a regression algorithm.

Customer segmentation
Identify groups of customers with similar profiles using a clustering algorithm.

GitHub labeler
Suggest the GitHub label for new issues using a multi-class classification algorithm.

Fraud detection
Detect fraudulent credit card transactions using a binary classification algorithm.

Spam detection
Flag text messages as spam using a binary classification algorithm.

Image classification
Classify images (e.g. broccoli vs pizza) using a TensorFlow deep learning algorithm.

Sales forecasting
Forecast future sales for products using a regression algorithm.

dot.net/ml

github.com/dotnet/machinelearning-samples

RELEASED

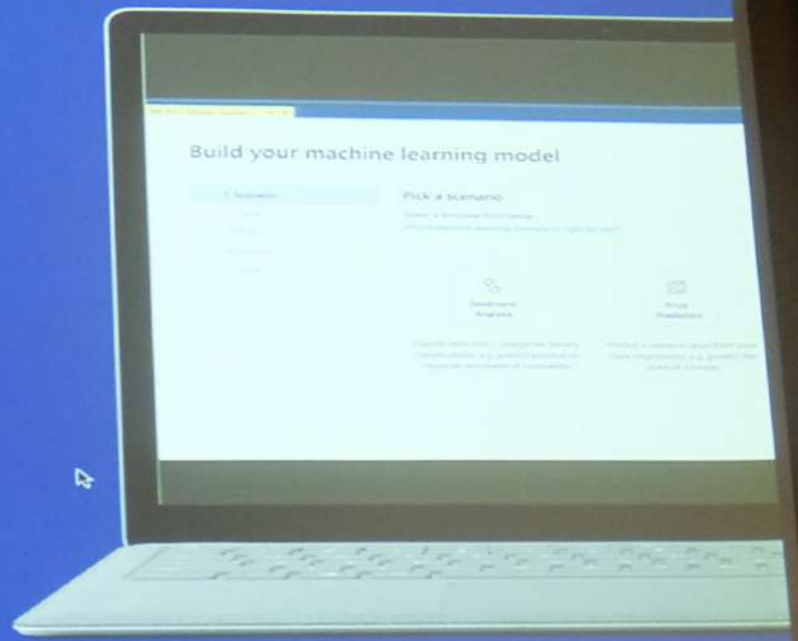
ML.NET Model Builder

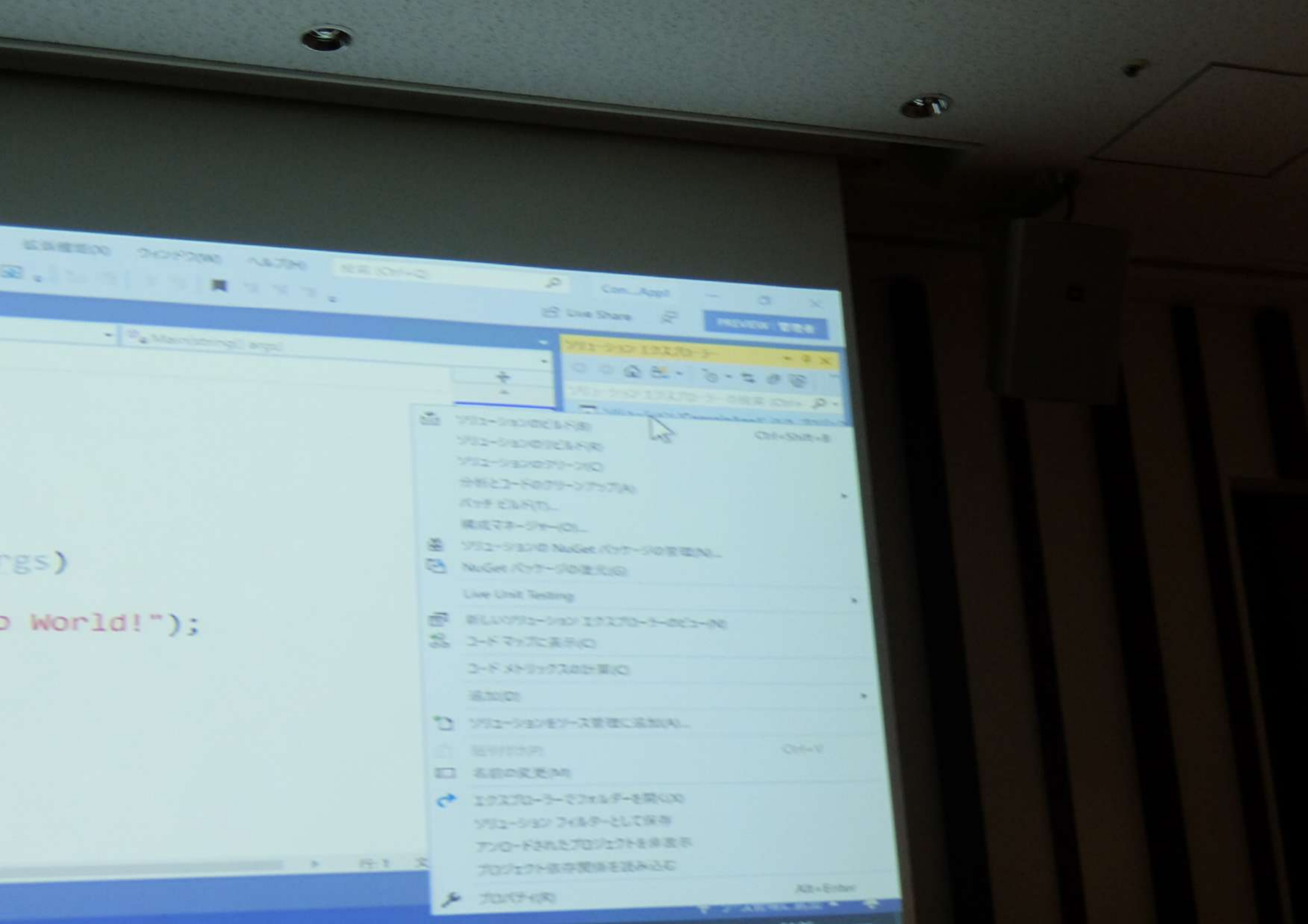
Simple UI tool for developers

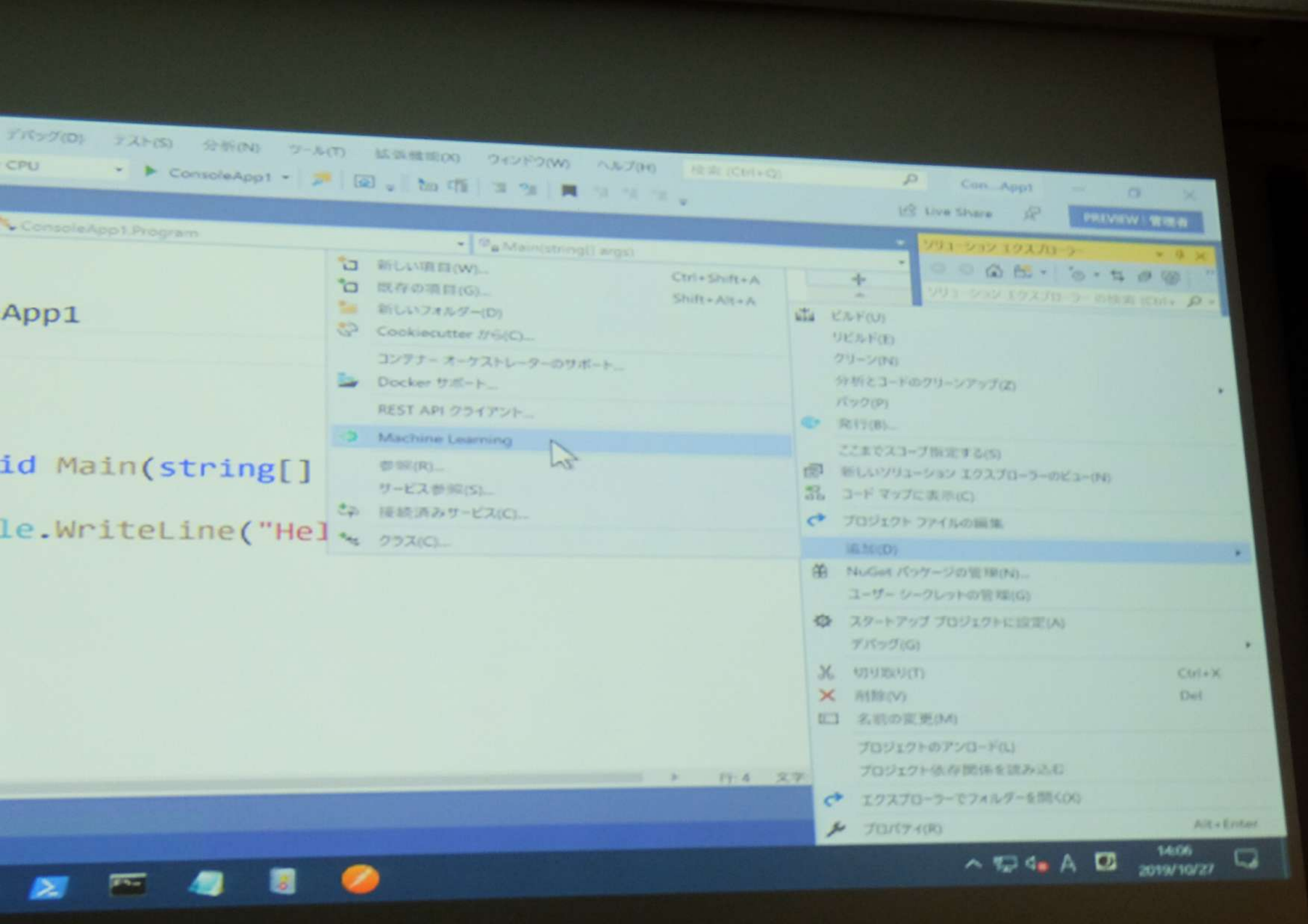
Build custom ML models easily with AutoML

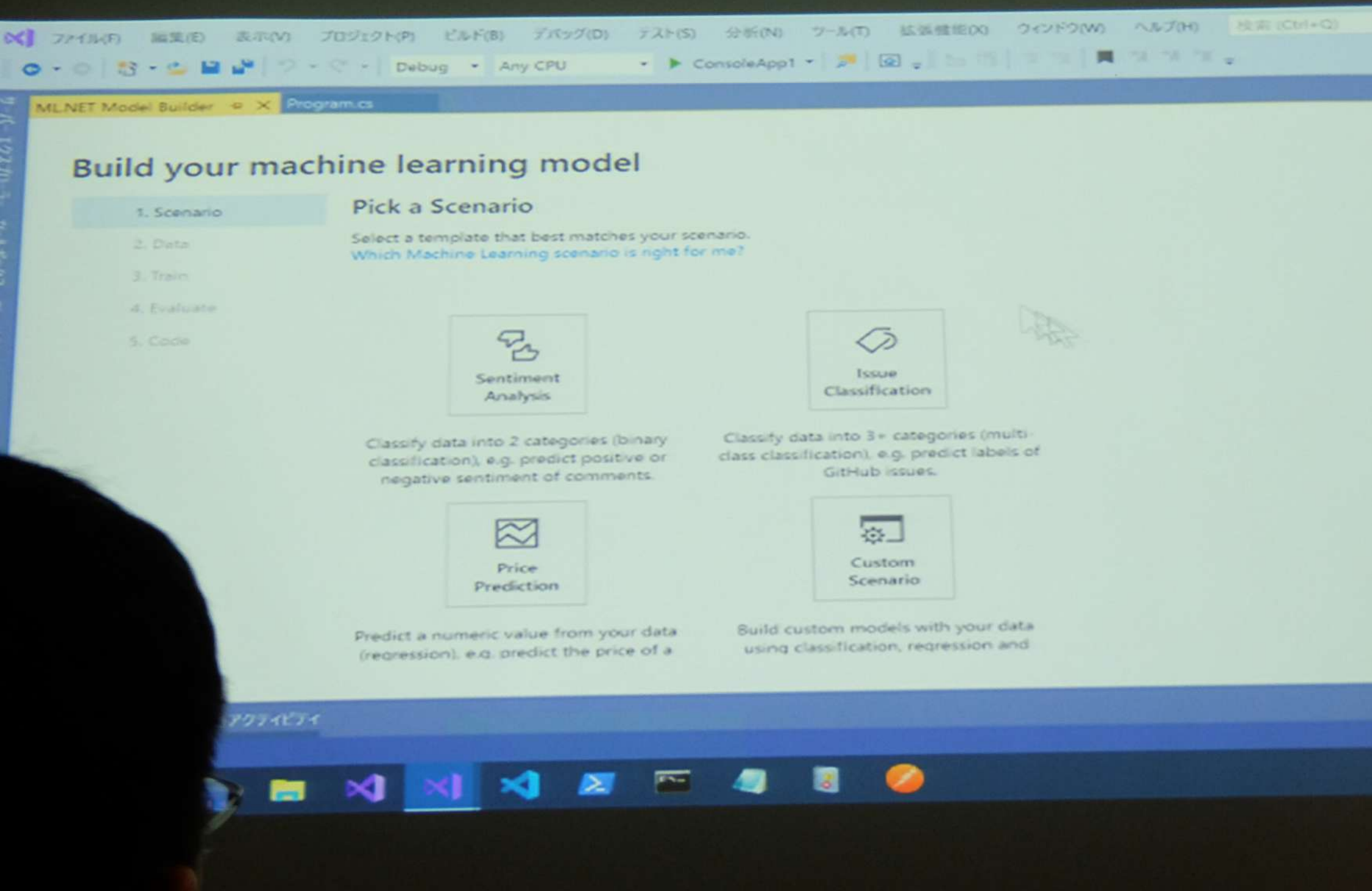
Generate code for training & consumption

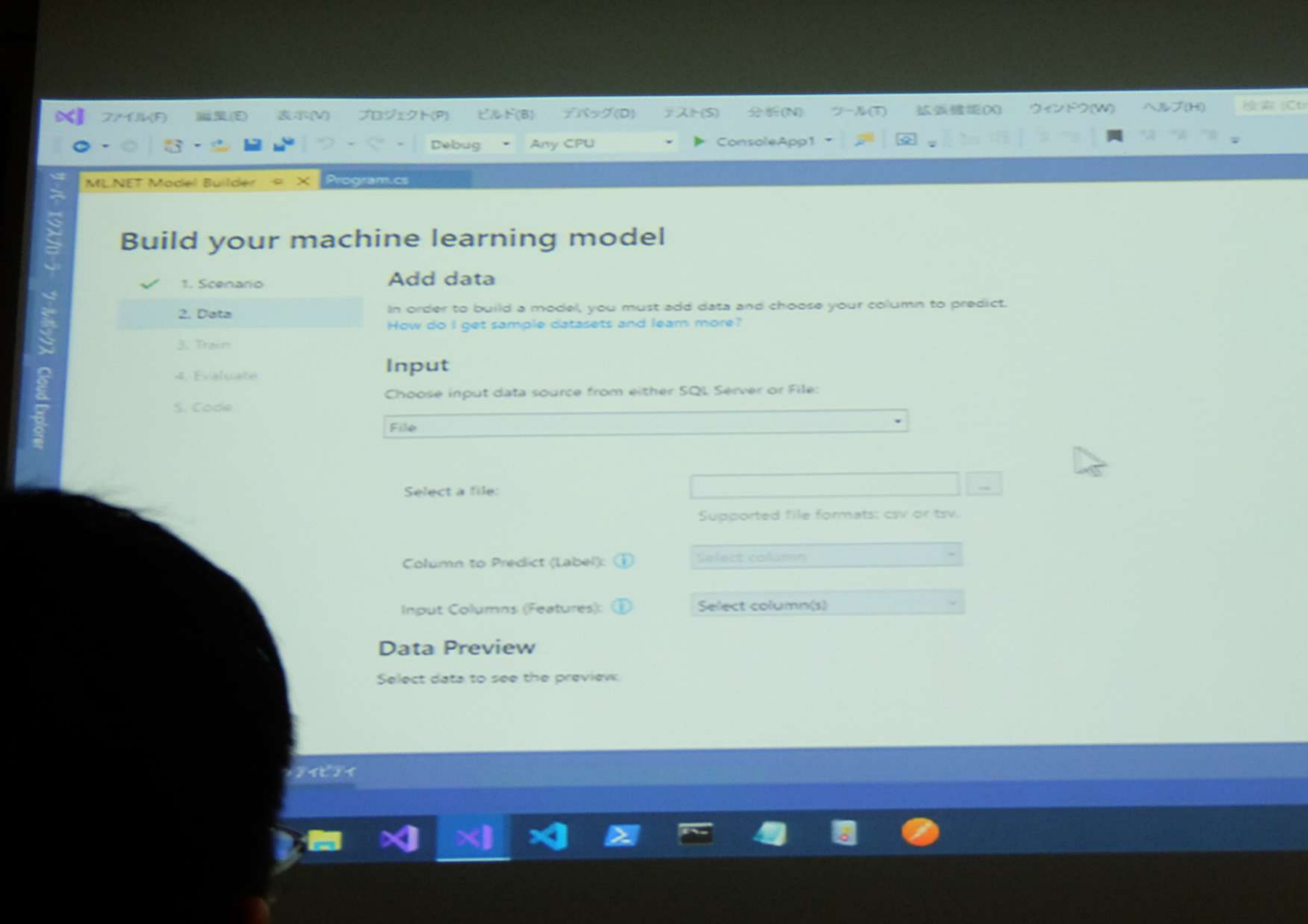
dot.net/ml

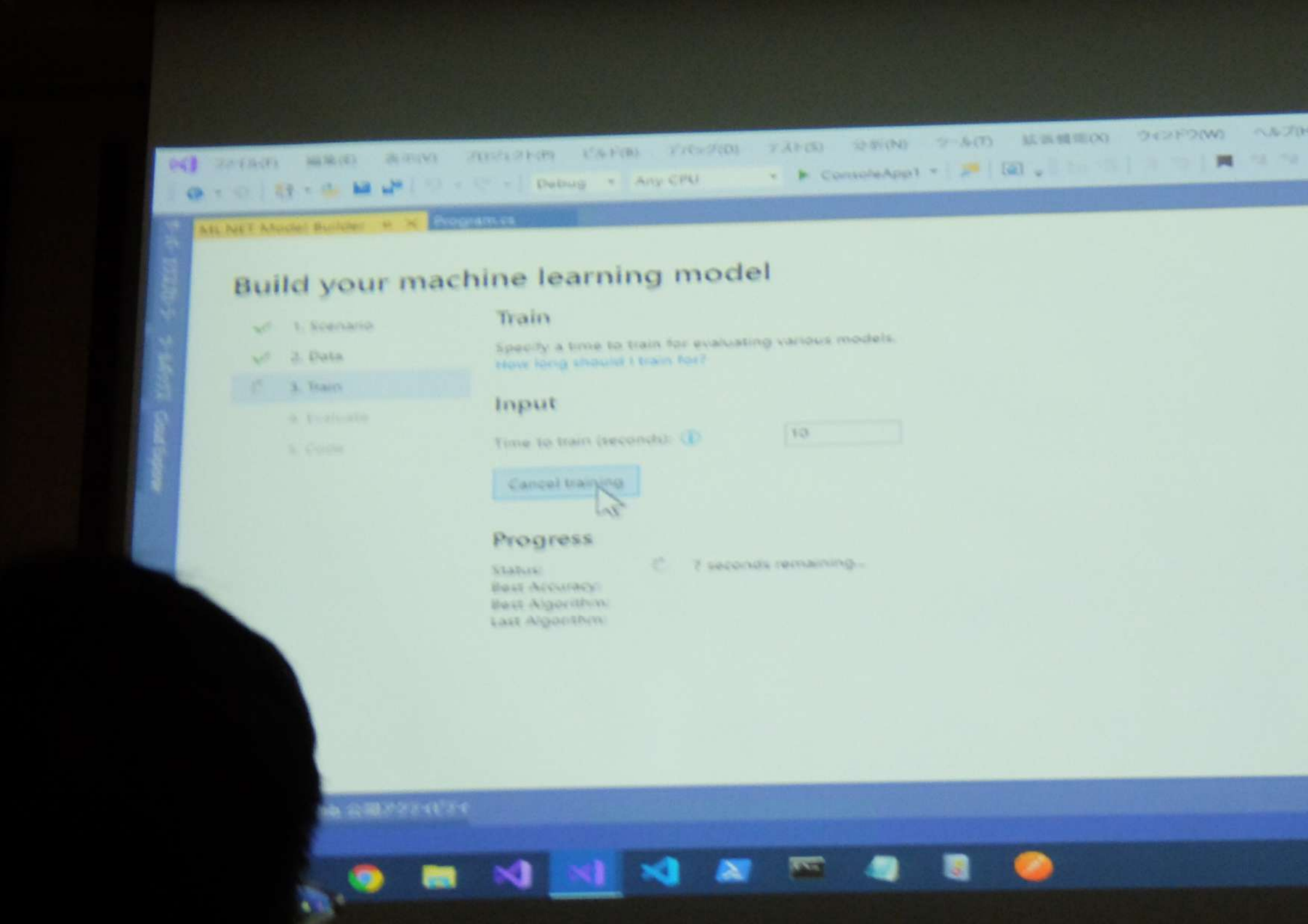


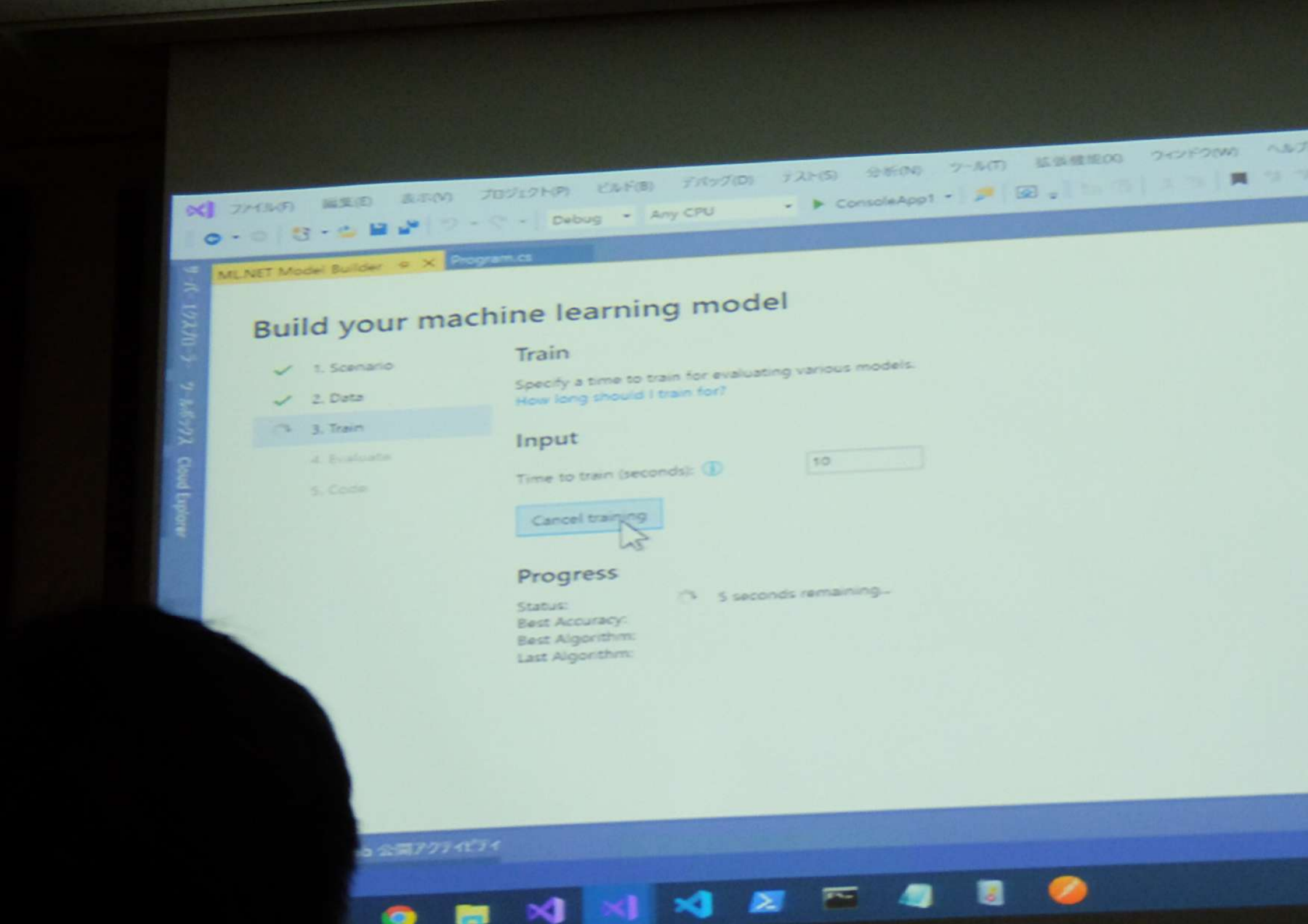

rgs)
o World!");


デバッグ(D)
テスト(S)
分析(N)
ツール(T)
拡張機能(X)
ウィンドウ(W)
ヘルプ(H)
CPU
ConsoleApp1
Live Share
PREVIEW
ConsoleApp1.Program
Main(string[] args)
App1
id Main(string[]
le.WriteLine("He
新しい項目(W)... Ctrl+Shift+A
既存の項目(G)... Shift+Alt+A
新しいフォルダー(D)
Cookiecutter から(C)...
コンテナー オーケストレーターのサポート...
Docker サポート...
REST API クライアント...
Machine Learning
参照(R)...
サービス参照(S)...
接続済みサービス(C)...
クラス(C)...
ビルド(U)
リビルド(E)
クリーン(N)
パック(P)
NuGet パッケージの管理(N)...
スタートアップ プロジェクトに設定(A)
デバッグ(G)
切り取り(T) Ctrl+X
削除(V) Del
名前の変更(M)
プロジェクトのアンロード(L)
エクスプローラーでフォルダーを開く(X)
プロパティ(R) Alt+Enter
2019/10/27


ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) テスト(S) 分析(N) ツール(T) 拡張機能(X) ウィンドウ(W) ヘルプ(H)
Debug
Any CPU
ConsoleApp1
MLNET Model Builder
Program.cs
Build your machine learning model
1. Scenario
2. Data
3. Train
4. Evaluate
5. Code
Pick a Scenario
Select a template that best matches your scenario.
Which Machine Learning scenario is right for me?
Sentiment Analysis
Classify data into 2 categories (binary classification), e.g. predict positive or negative sentiment of comments.
Issue Classification
Classify data into 3+ categories (multi-class classification), e.g. predict labels of GitHub issues.
Price Prediction
Predict a numeric value from your data (regression), e.g. predict the price of a
Custom Scenario
Build custom models with your data using classification, regression and


ファイル(F) 編集(E) 表示(V) プロジェクト(P) ビルド(B) デバッグ(D) テスト(S) 分析(N) ツール(T) 拡張機能(X) ウィンドウ(W) ヘルプ(H)
Debug
Any CPU
ConsoleApp1
ML.NET Model Builder
Program.cs
Build your machine learning model
1. Scenario
2. Data
3. Train
4. Evaluate
5. Code
Add data
In order to build a model, you must add data and choose your column to predict.
How do I get sample datasets and learn more?
Input
Choose input data source from either SQL Server or File:
File
Select a file:
Supported file formats: csv or tsv.
Column to Predict (Label):
Select column
Input Columns (Features):
Select column(s)
Data Preview
Select data to see the preview.
Cloud Explorer

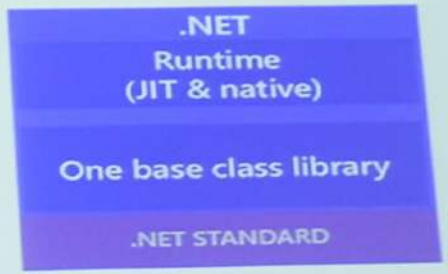

Debug
Any CPU
ConsoleApp1
ML.NET Model Builder
Program.cs
Build your machine learning model
1. Scenario
2. Data
3. Train
4. Evaluate
5. Code
Train
Specify a time to train for evaluating various models.
How long should I train for?
Input
Time to train (seconds):
10
Cancel training
Progress
Status:
Best Accuracy:
Best Algorithm:
Last Algorithm:
7 seconds remaining...


Debug
Any CPU
ConsoleApp1
ML.NET Model Builder
Program.cs
Build your machine learning model
1. Scenario
2. Data
3. Train
4. Evaluate
5. Code
Train
Specify a time to train for evaluating various models.
How long should I train for?
Input
Time to train (seconds):
10
Cancel training
Progress
5 seconds remaining...
Status:
Best Accuracy:
Best Algorithm:
Last Algorithm:


.NET Roadmap

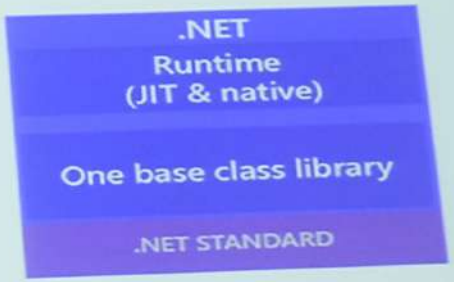
Introducing .NET 5

- すべての .NET Core / Xamarin アプリケーション モデルをサポート
- 統一プラットフォームへ向けた .NET Core の進化
  - 一つの Base Class Library (BCL) 実装 + .NET Standard
  - 一つのツール チェーン (SDK スタイル プロジェクト)
  - Just-in-Time (JIT) とネイティブの実行モデルをサポート
  - Java, Swift との相互運用性 (モバイル開発 – Xamarin)
- .NET 5 に含まれないアプリケーション モデル
  - ASP.NET Web Forms
    - 代替は Blazor (移行ガイド ドキュメント公開予定)
  - WCF (Windows Communication Foundation)
    - 代替は gRPC for WCF server and remoting (移行ガイド ドキュメント公開予定)
  - WF (Windows Workflow Foundation)
    - 代替は Open source core workflow for Windows workflow (WF): https://github.com/UiPath/corewf

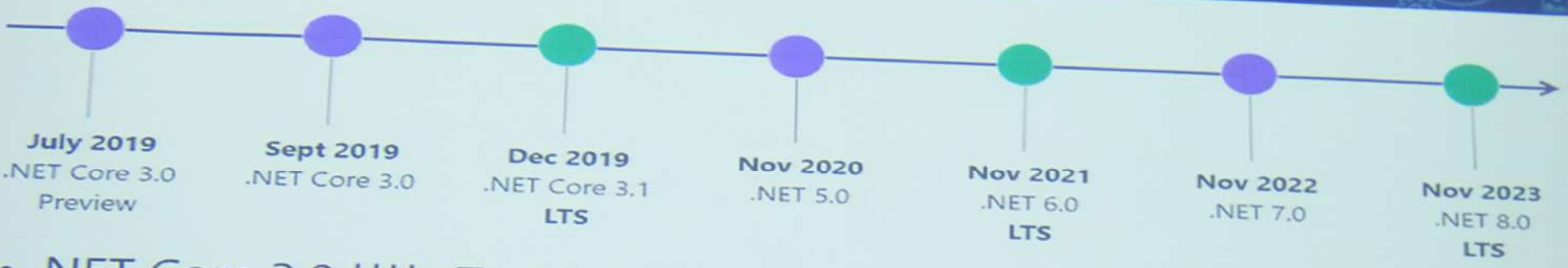


Introducing .NET 5

- すべての .NET Core / Xamarin アプリケーション モデルをサポート
- 統一プラットフォームへ向けた .NET Core の進化
  - 一つの Base Class Library (BCL) 実装 + .NET Standard
  - 一つのツール チェーン (SDK スタイル プロジェクト)
  - Just-in-Time (JIT) とネイティブの実行モデルをサポート
  - Java, Swift との相互運用性 (モバイル開発 – Xamarin)
- .NET 5 に含まれないアプリケーション モデル
  - ASP.NET Web Forms
    - 代替は Blazor (移行ガイド ドキュメント公開予定)
  - WCF (Windows Communication Foundation)
    - 代替は gRPC for WCF server and remoting (移行ガイド ドキュメント公開予定)
  - WF (Windows Workflow Foundation)
    - 代替は Open source core workflow for Windows workflow (WF): https://github.com/UiPath/corewf

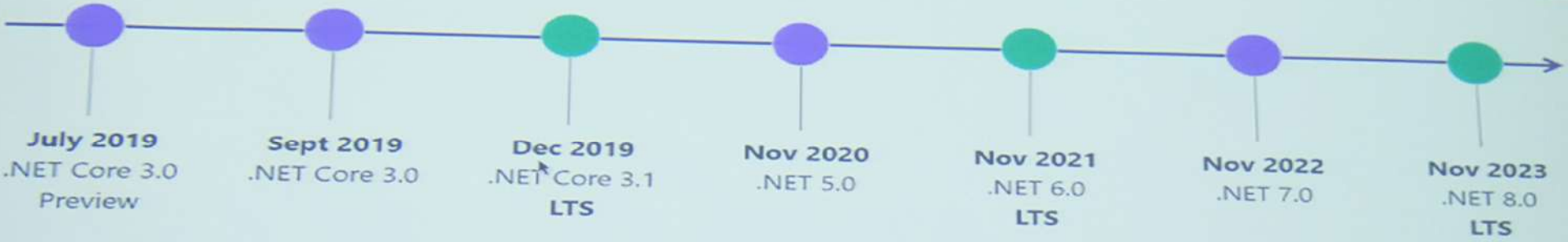


.NET スケジュール

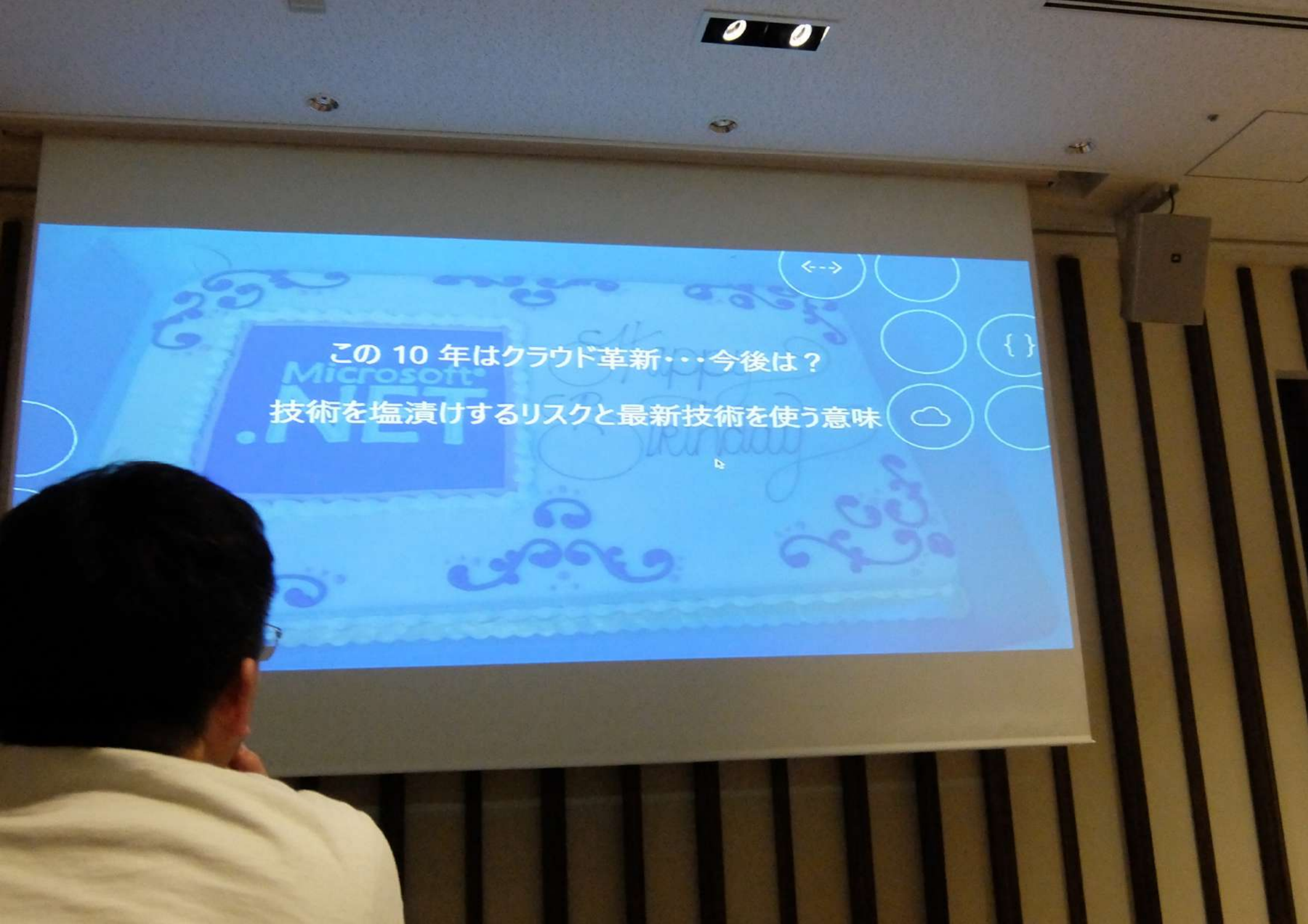


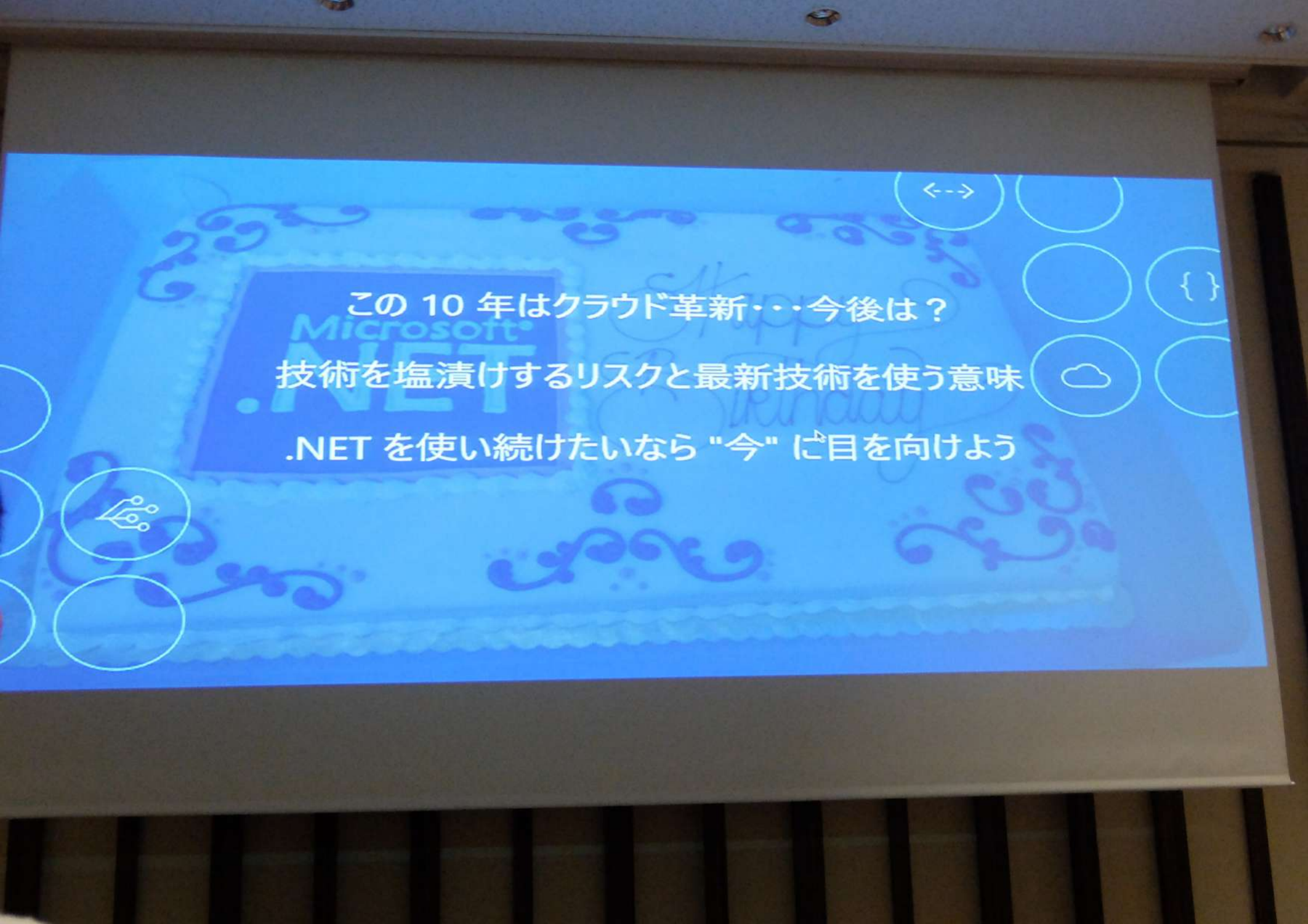
- .NET Core 3.0 リリース : 2019 年 9 月 23 日
- .NET Core 3.1 : Long Term Support (LTS)
- .NET 5.0 リリース : 2020 年 11 月
- 毎年 11 月にメジャー リリース, LTS は奇数年リリース
- 予定リリース以外に必要であればマイナーリリースあり

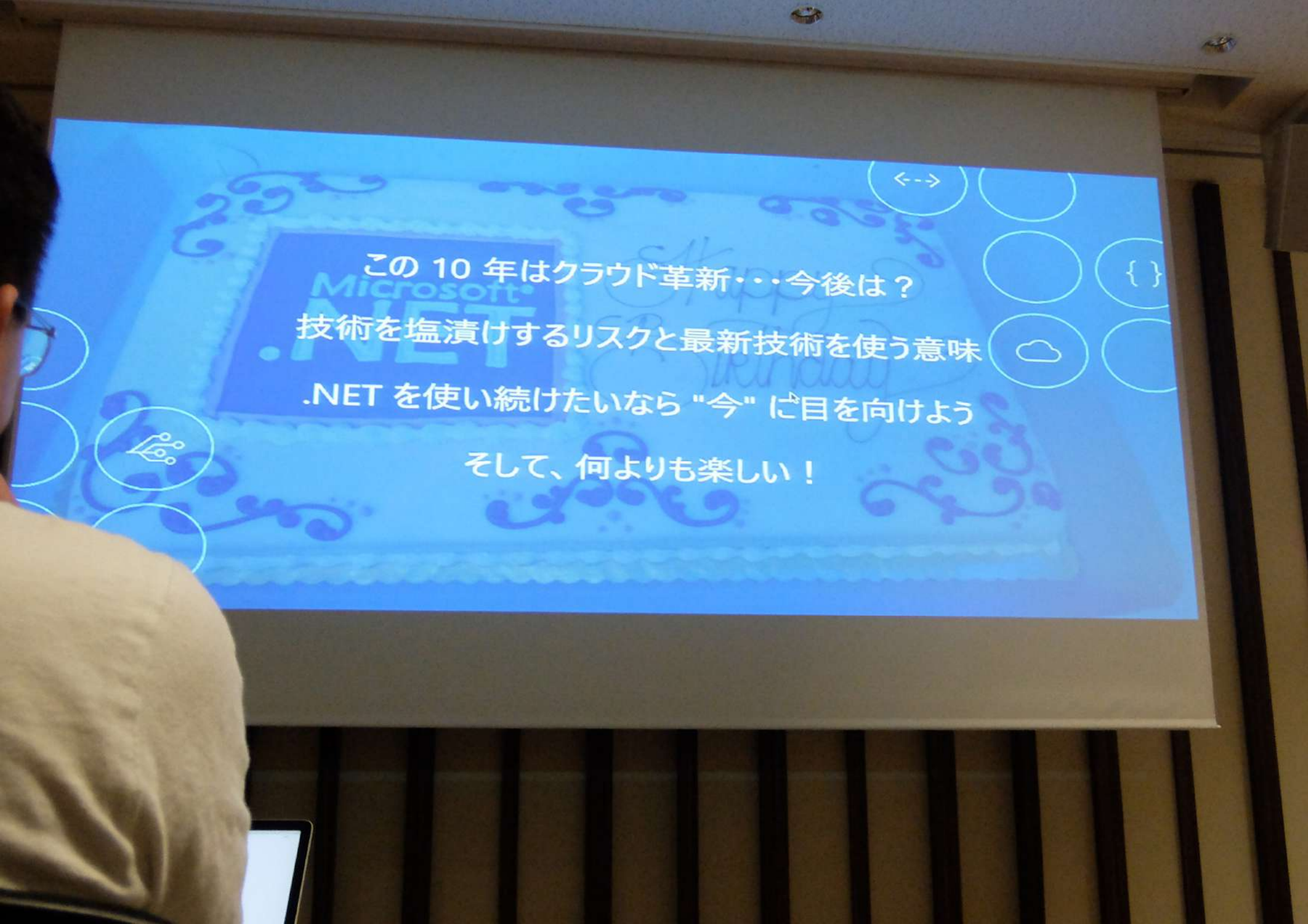
.NET スケジュール



- .NET Core 3.0 リリース : 2019 年 9 月 23 日
- .NET Core 3.1 : Long Term Support (LTS)
- .NET 5.0 リリース : 2020 年 11 月
- 毎年 11 月にメジャー リリース, LTS は奇数年リリース
- 予定リリース以外に必要であればマイナーリリースあり


この 10 年はクラウド革新・・・今後は？
技術を塩漬けするリスクと最新技術を使う意味


この 10 年はクラウド革新・・・今後は？
技術を塩漬けするリスクと最新技術を使う意味
.NET を使い続けたいなら "今" に目を向けよう


この 10 年はクラウド革新・・・今後は？
技術を塩漬けするリスクと最新技術を使う意味
.NET を使い続けたいなら "今" に目を向けよう
そして、何よりも楽しい！