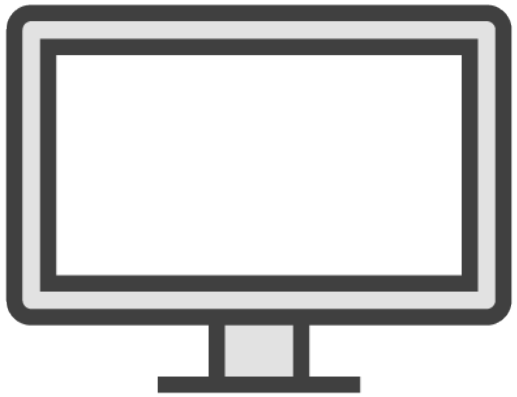
Rebuilding Web Forms Applications in MVC

PUTTING WEB FORMS AND MVC IN PERSPECTIVE

Alex Wolf

www.alexwolfthoughts.com

Rebuilding and Improving



Old Application

New Application
New Requirements
New Framework

Transitioning Between Web Forms and MVC

High Level Concepts

Hey, we have things in common!

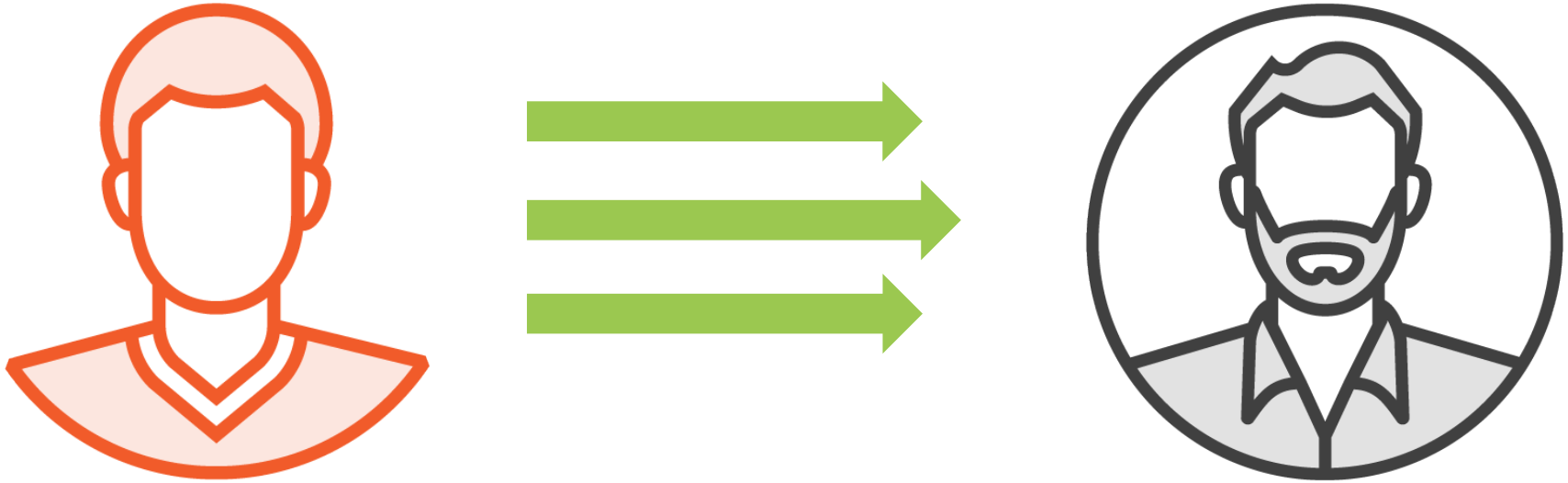
Implementation Details

We're pretty different...

Moving Forward...

Are You in the Right Place?

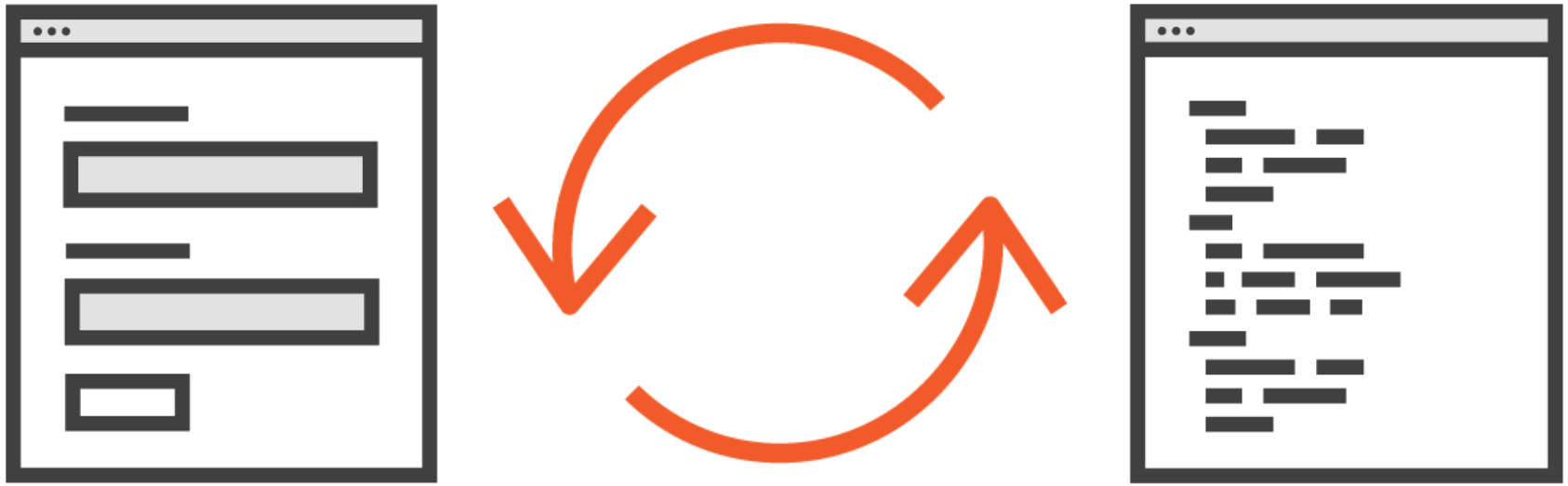
The Transitional Developer



Web Forms Developer

MVC Developer

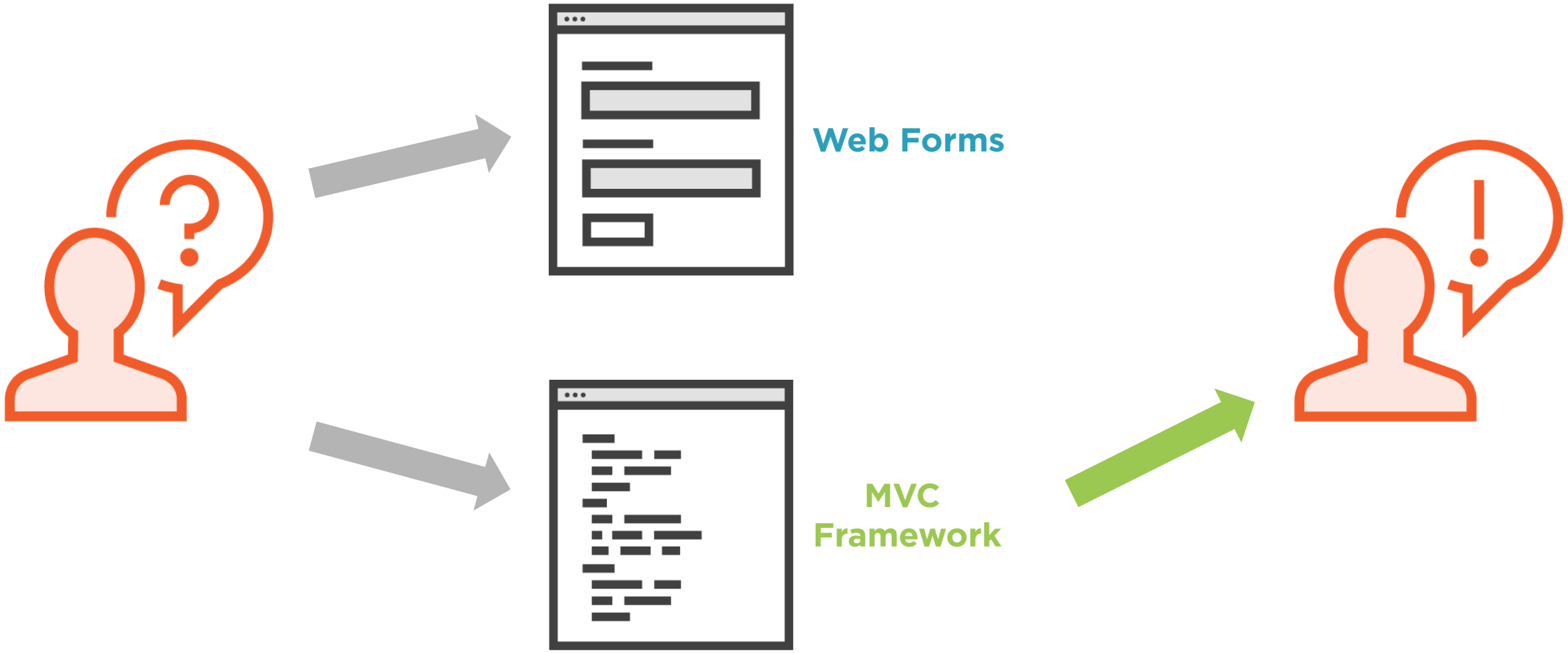
The Reverse Engineer



Web Forms Concepts

MVC Concepts

The Empowered Decision Maker



Managing Expectations

Need to Know

General Web Development

HTML

CSS

HTTP Requests

ASP.NET

General Understanding of the Platform

Some Web Forms or Some MVC

Nice to Know

Other Technologies

Entity Framework

jQuery Basics

Creating Context

Understanding Web Forms

Understanding Web Forms

Built on ASP.NET

- HTTP Modules and Handlers
- Security and User Roles
- Session
- Caching

Markup Generation

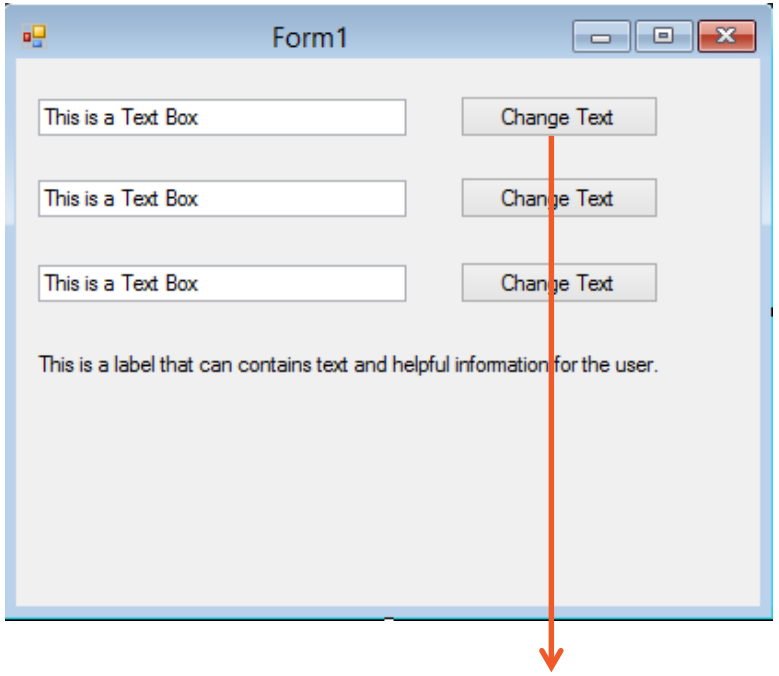
Illusion of State

Code Separation

Reusable Components

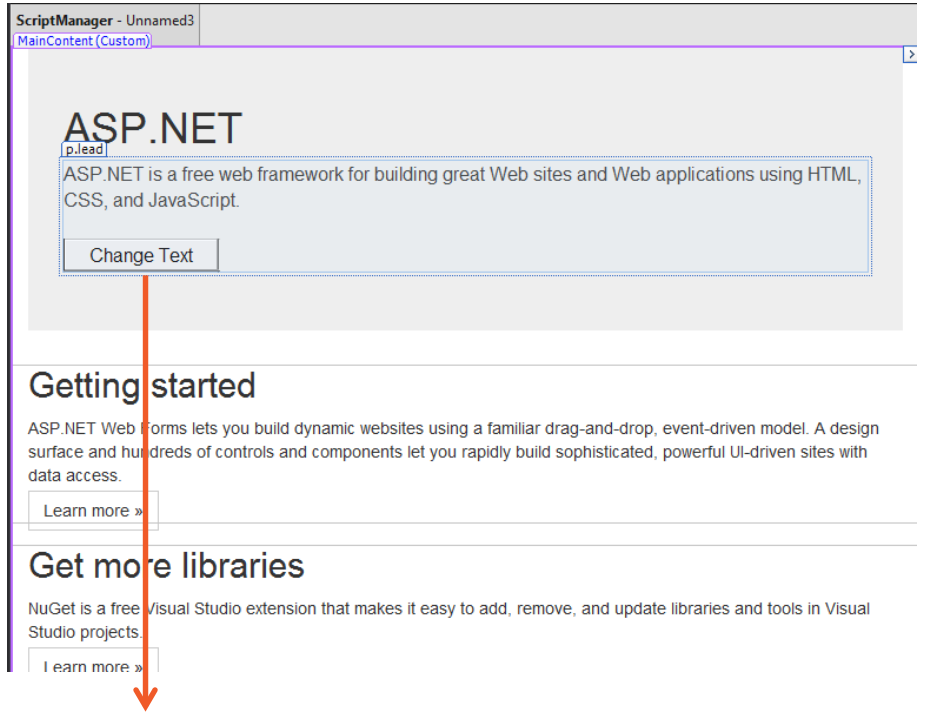
From Windows Forms to Web Forms

Windows Forms




```
protected void Btn1_Click(object sender, EventArgs args)
{
        textBox1.Text = "This is a Windows Form";
}
```

Web Forms




```
protected void Btn1_Click(object sender, EventArgs args)
{
        label1.Text = "This is a Web Form";
}
```

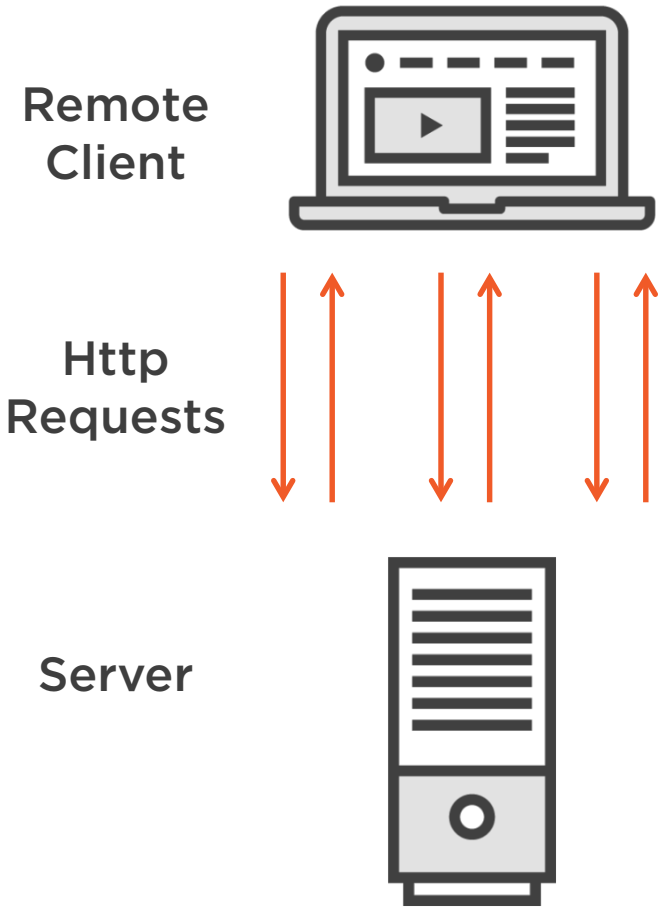
```
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="Data
Source=(localdb)\v11.0;Initial Catalog=CypherMVC;Integrated Security=True"
OnSelecting="SqlDataSource1_Selecting" ProviderName="System.Data.SqlClient“
SelectCommand="SELECT Subject, Author, Created FROM Messages"></asp:SqlDataSource>

<asp:ListView ID="ItemList" runat="server"
OnSelectedIndexChanged="ItemList_SelectedIndexChanged"
DataSourceID="SqlDataSource1"></asp:ListView>
```

Web Forms Control Mark Up

Often troubling, both before and after rendering

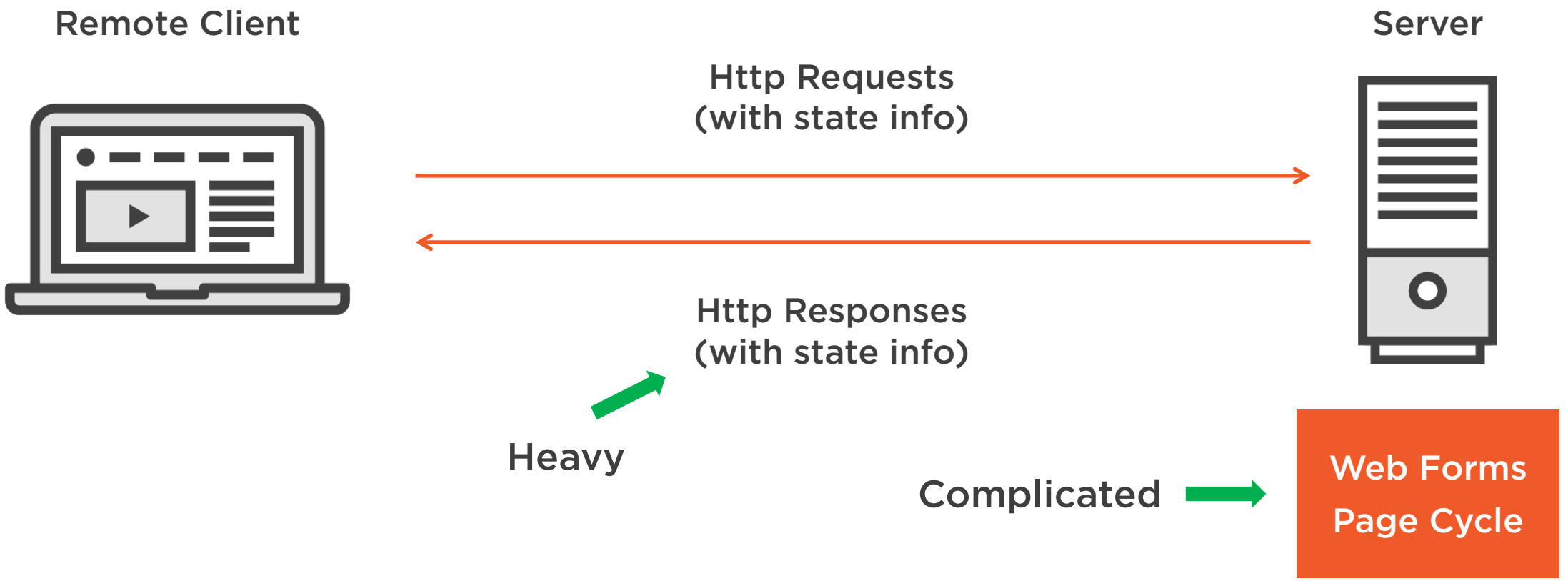
From Windows Forms to Web Forms

Web Applications



Desktop Client Applications

Enhanced Web Application Requests



Separation of Concerns

Classic Applications
("spaghetti" code)

```
<div class="row section-1">
(server side logic....)
</div>

<div class="row section-2">
(server side logic....)
</div>

<div class="row section-3">
(server side logic....)
</div>
```

Modern Applications
(separated code)

Client Side Code

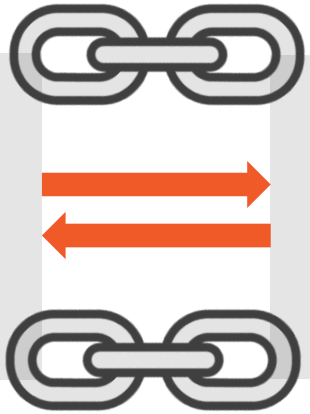
```
<div class="row section-1"></div>

<div class="row section-2"></div>
```


```
(server side logic....)
```

Tightly Coupled Components

Client Side Code
(Home.aspx)

```
<div class="row section-1"></div>

<div class="row section-2”></div>
```



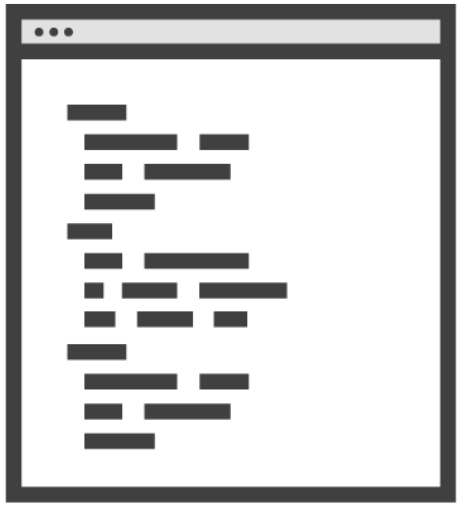
Server Side Code
(Home.aspx.cs)

(server side logic....)

Action Oriented Design

Hey, I need to check out from my mobile app!

Okay, here's a JSON response



I'm on the full website....

You can have full HTML

Other Considerations

Unit Testing

Dependency Injection

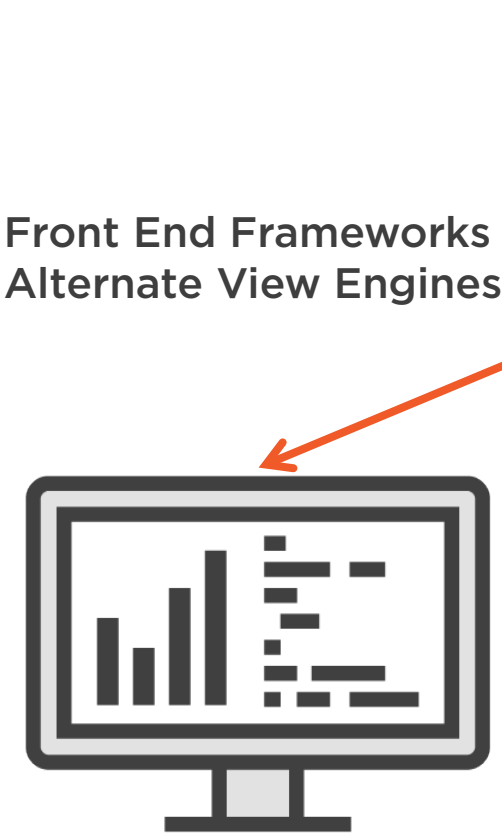
Flexible Web Service End Points

A New Destination

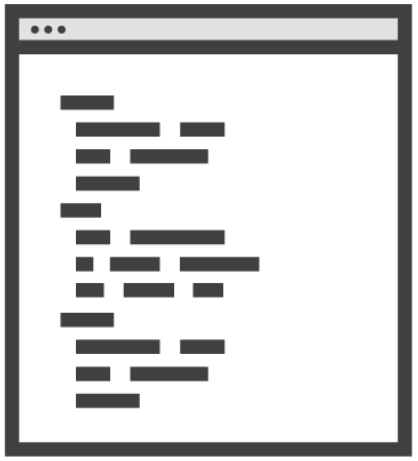
Introducing MVC

The MVC Pattern

- ✓ Front End Frameworks
- ✓ Alternate View Engines



View



Controller

- ✓ Service Layers
- ✓ Repository Layers

Model

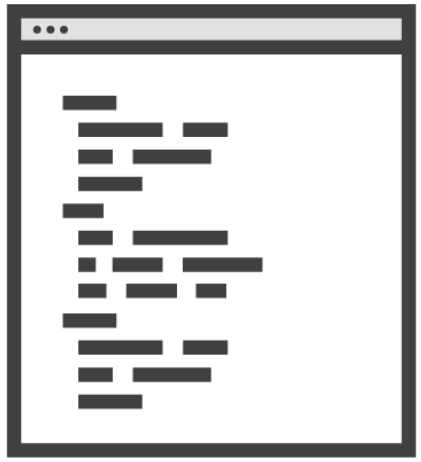
Embracing the Nature of HTTP

Remote Client

Http Requests
(No View State)

Http Responses
(No View State)

Controller Methods



Flexibility Through Extensibility

Standard Behavior

Use the Default Conventions

Tweaked Behavior

Extend the Existing Components

Fully Custom Behavior

Rewrite and Replace the Provided Classes

Controlling Your Markup

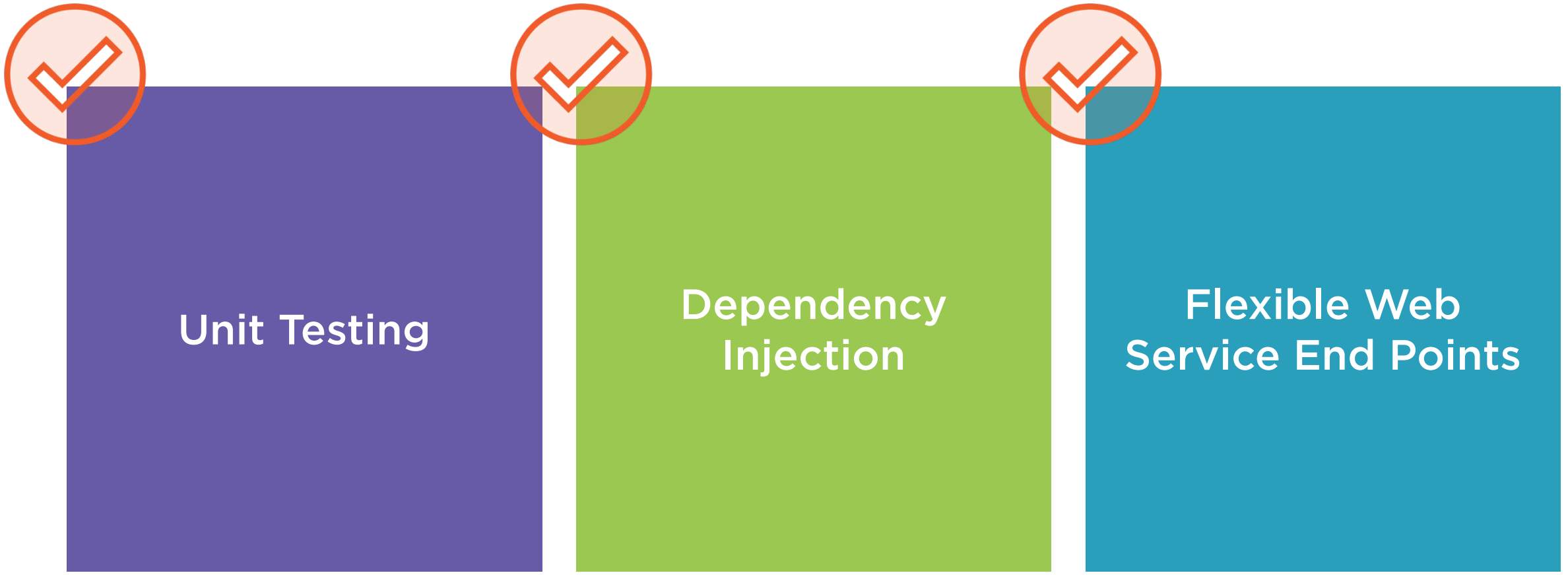
Razor Code		Generated Markup
@Html.TextBoxFor(p => p.Age)	→	<input type="text" name="Age" />
@Html.TextAreaFor(p => p.Comments)	→	<textarea name="Comments"> </textarea>
@Html.ActionLink("Home", "Index", "Go Home")	→	Go Home

```
public class HomeController {

    public ActionResult Index()
    {
        return View();
    }


    public ActionResult GetFeed()
    {
        var feed = db.GetFeed()
        return Json(feed);
    }
}
```

- Responds to home page request with HTML

- Same controller responds with JSON feed

Embracing Modern Design Patterns



MVC and ASP.NET

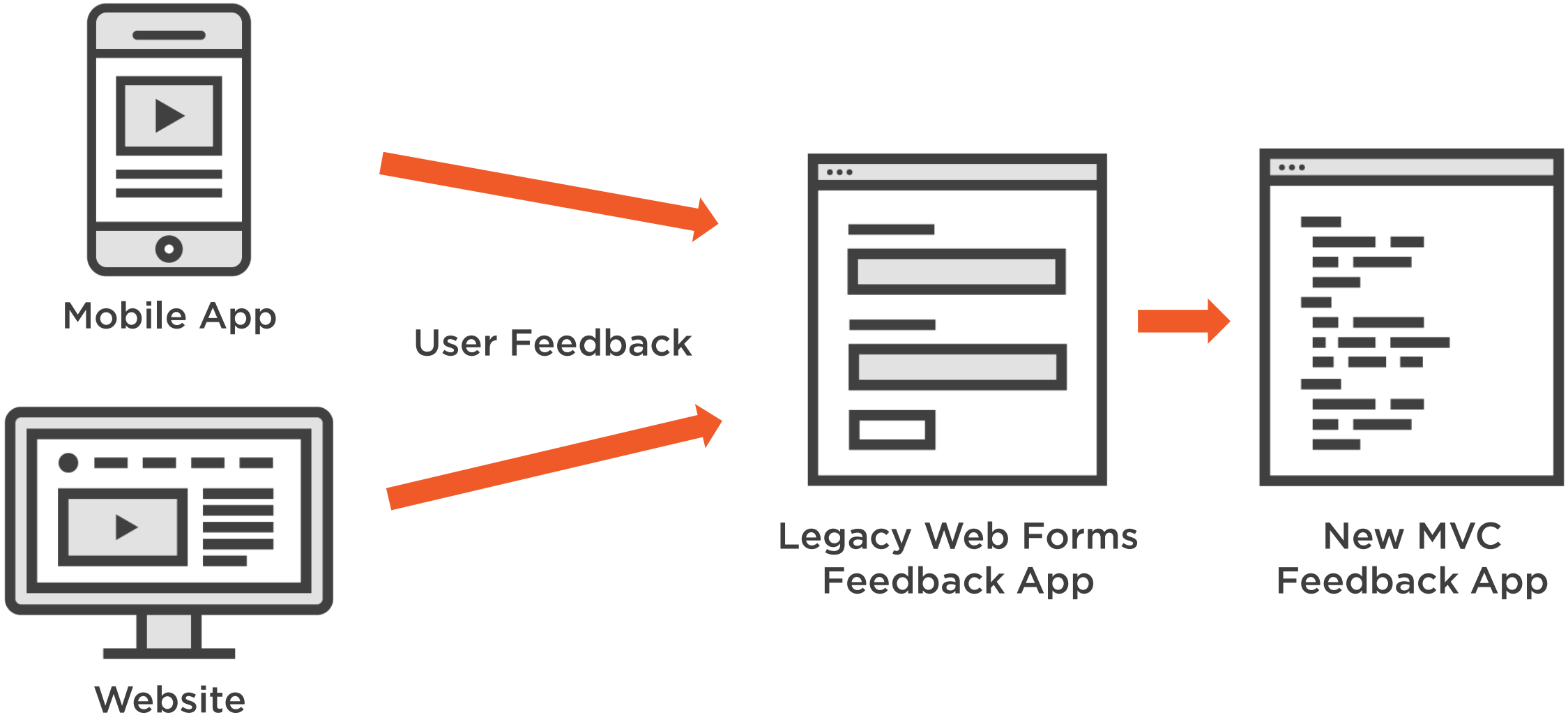
The Agenda

To-Do List

- Request Management
- Designing with Layouts and Views
- Forms and Model Binding
- Validating Form Data
- User Controls and Partial Views
- Handling Ajax and Service Calls
- Displaying and Accessing Data

Building for the Future

Responding to Feedback



Summary

Web Forms Offers Interesting but Challenging Features

- View State
- HTML and HTTP Abstractions
- Split Page Model
- Page and Form Validation

Both Web Forms and MVC are Built on ASP.NET

MVC is Powerful and Extensible

Built for Modern Design Patterns

MVC Embraces Nature of Web