Defensive Coding Techniques

Deborah Kurata
Consultant | Developer | Mentor

Level: Beginner/Intermediate

Deborah Kurata

- Independent Consultant | Developer | Mentor
  - Web (Angular), .NET
- Pluralsight Author
  - Angular with TypeScript
  - Angular Front to Back with Web API
  - Defensive Coding in C#
  - Object-Oriented Programming Fundamentals in C#
- Microsoft MVP

Session Materials & Sample Code

https://github.com/DeborahK/VSLive2015-NY

What is Defensive Coding?

… an approach to improve software and source code, in terms of:

- General $\boxed{\text{quality}}$ - Reducing the number of software bugs and problems.
- Making the source code $\boxed{\text{comprehensible}}$ - the source code should be readable and understandable so it is approved in a code audit.
- Making the software behave in a $\boxed{\text{predictable}}$ manner despite unexpected inputs or user actions.

- Wikipedia as of 2/12/15

Clean Code

Automated Code Testing

Validation + Exception Handling

Defensive Coding

Clean Code

- Improves Comprehension
- Simplifies Maintenance
- Reduces Bugs

Testable Code
+
Unit Tests

- Improves Quality
- Confirms Maintenance
- Reduces Bugs

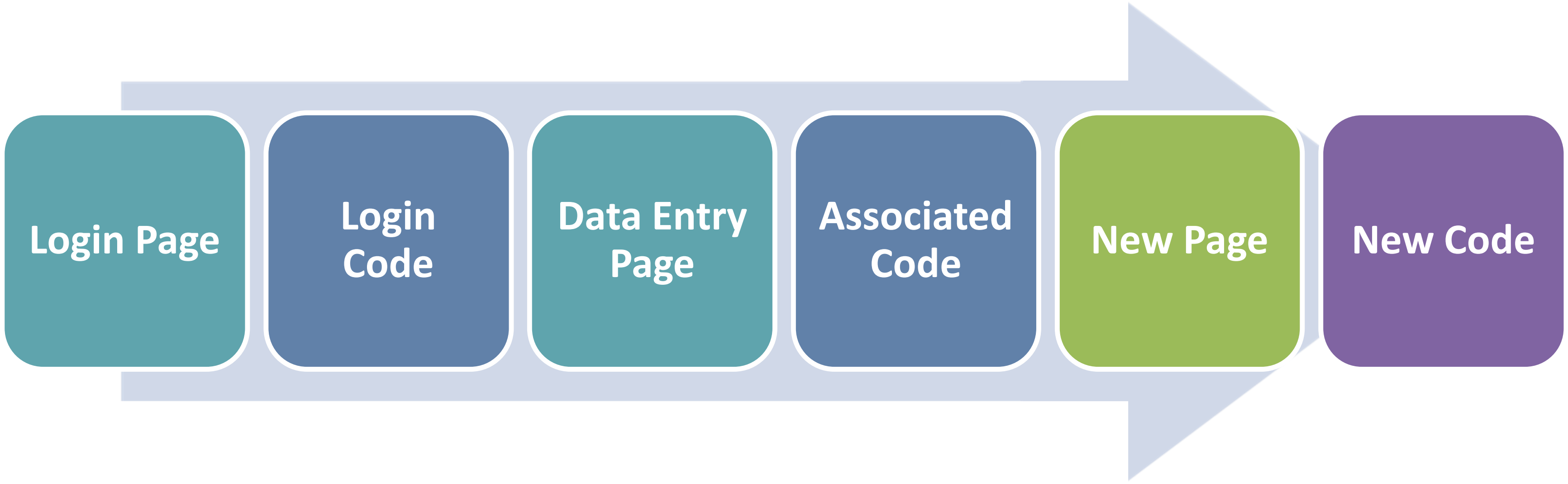
Validation
+
Exception Handling

- Improves Predictability
- More Consistent
- Reduces Bugs

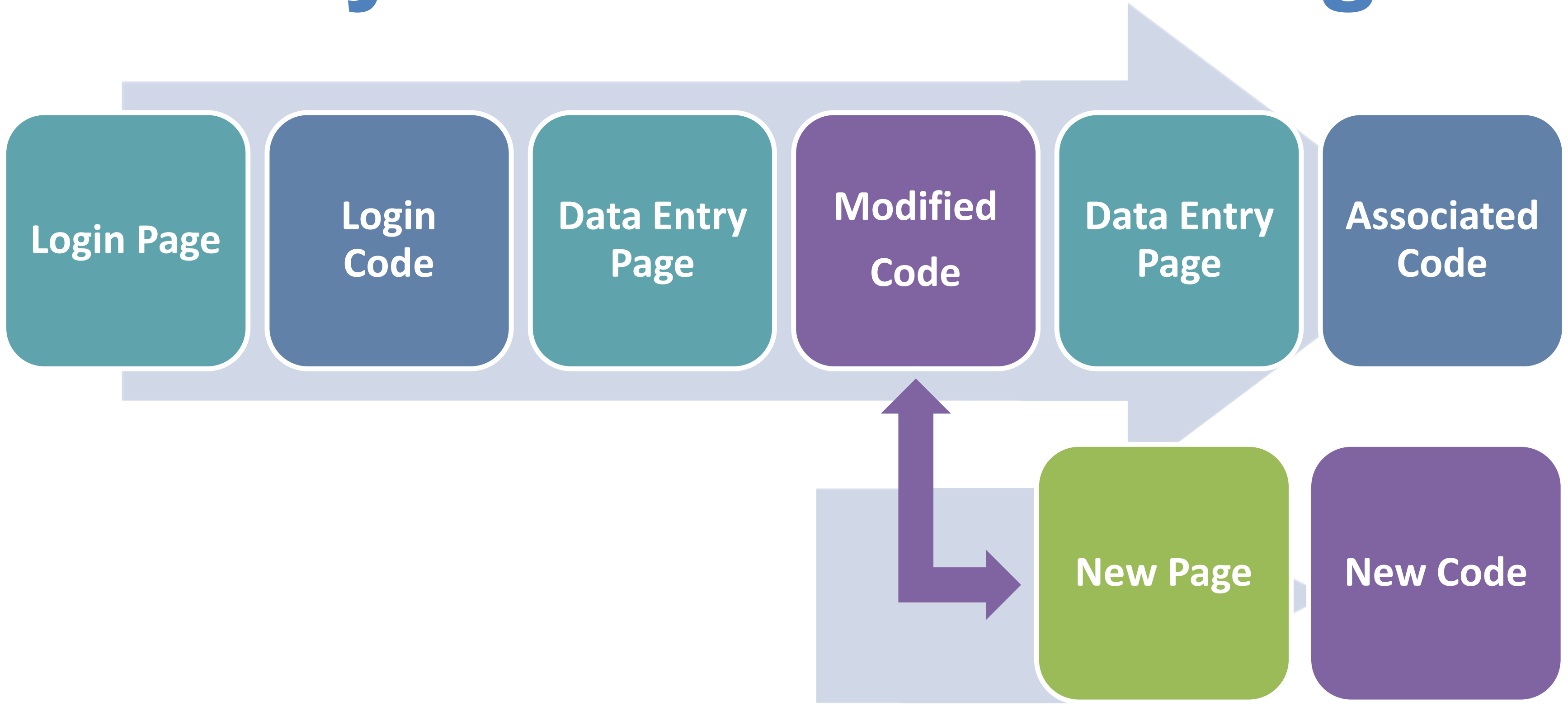
What Is Clean Code?

- ✓ Easy to read
- ✓ Clear intent
- ✓ Simple
- ✓ Minimal
- ✓ Thoughtful

Why Automated Testing?



Why Automated Testing?



Automated Code Testing

- ✓ Repeatable
- ✓ Built Into Visual Studio
- ✓ Arrange
- ✓ Act
- ✓ Assert

Validation: Trust

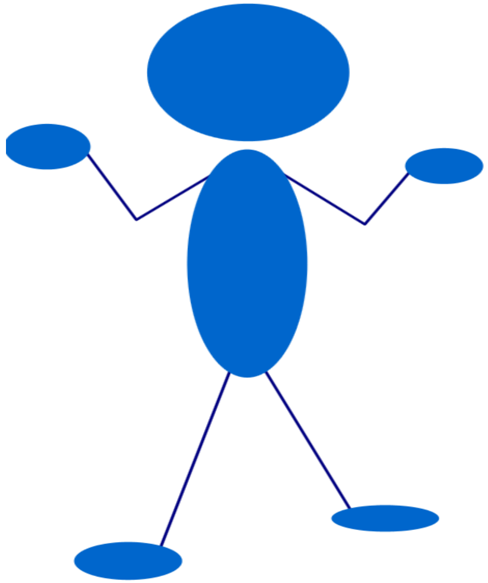
Contract

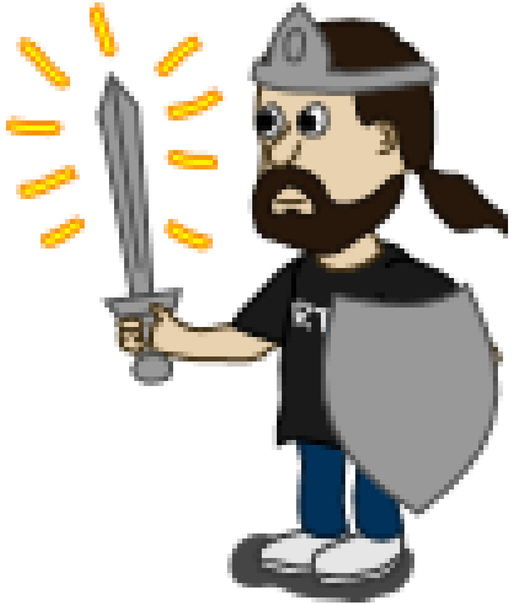
- Parameters
- Return Type
- Exceptions

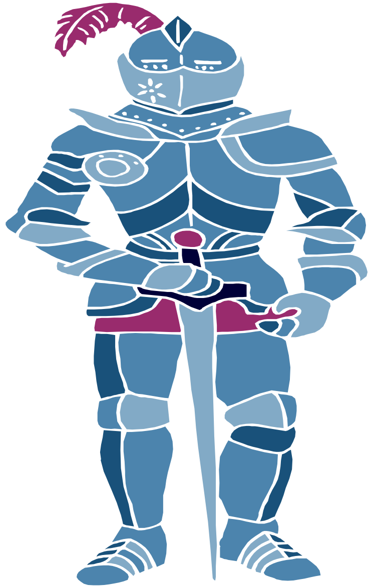
Verify

- Parameters
- Return Type
- Data
- Exceptions

Clean yet Protected







Defensive Coding

Clean Code

- Improves Comprehension
- Simplifies Maintenance
- Reduces Bugs

Testable Code + Unit Tests

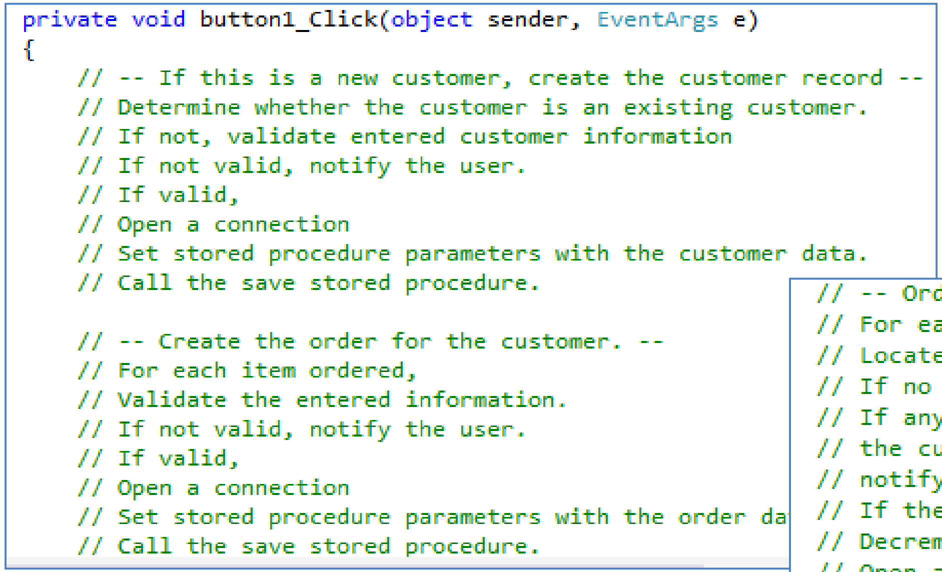
- Improves Quality
- Confirms Maintenance
- Reduces Bugs

Validation + Exception Handling

- Improves Predictability
- More Consistent
- Reduces Bugs

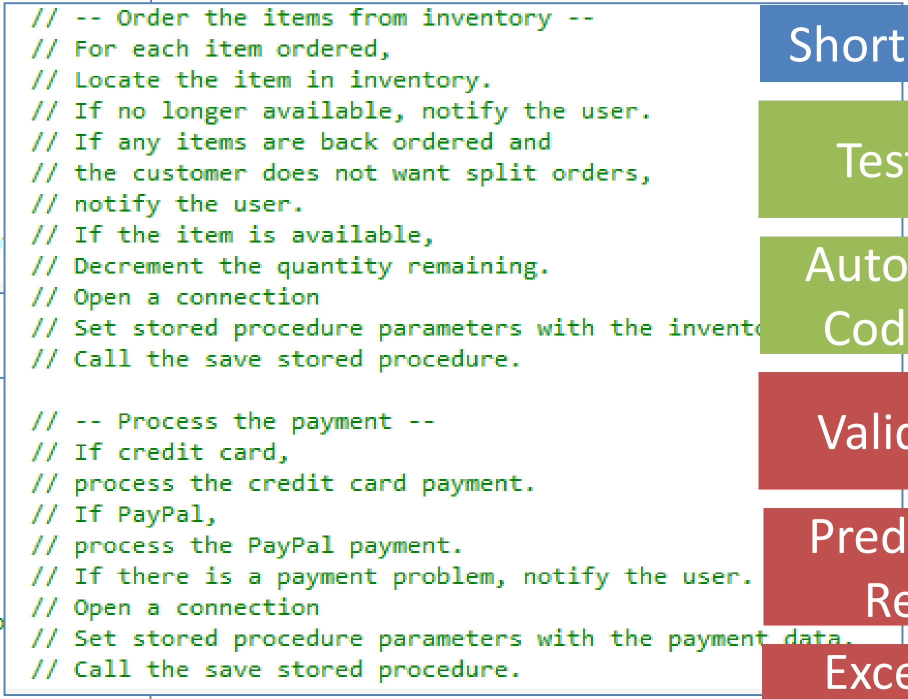
Considerations

Clean Code	Testable Code + Unit Tests	Validation + Exception Handling
Clear Purpose	Testable	Validation
Good Name	Automated Code Test	Predictable Result
Focused Code		Exception Handling
Short Length		



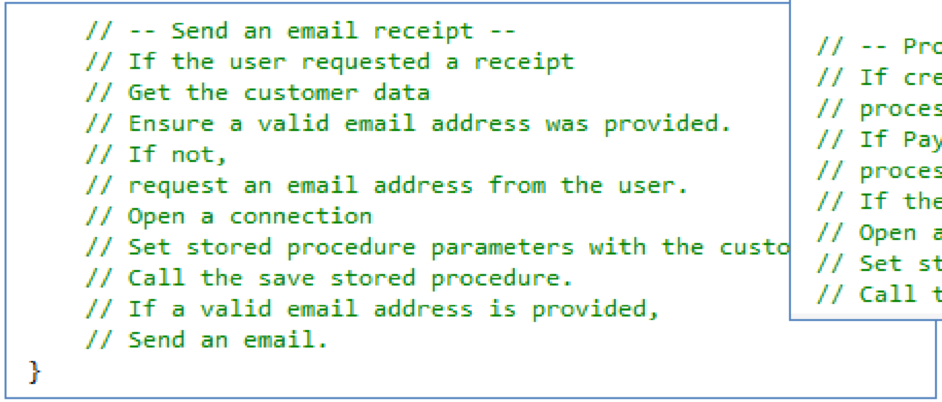
```
private void button1_Click(object sender, EventArgs e)
{
    // -- If this is a new customer, create the customer record --
    // Determine whether the customer is an existing customer.
    // If not, validate entered customer information
    // If not valid, notify the user.
    // If valid,
    // Open a connection
    // Set stored procedure parameters with the customer data.
    // Call the save stored procedure.

    // -- Create the order for the customer. --
    // For each item ordered,
    // Validate the entered information.
    // If not valid, notify the user.
    // If valid,
    // Open a connection
    // Set stored procedure parameters with the order da
    // Call the save stored procedure.
```



```
// -- Order the items from inventory --
// For each item ordered,
// Locate the item in inventory.
// If no longer available, notify the user.
// If any items are back ordered and
// the customer does not want split orders,
// notify the user.
// If the item is available,
// Decrement the quantity remaining.
// Open a connection
// Set stored procedure parameters with the invento
// Call the save stored procedure.

// -- Process the payment --
// If credit card,
// process the credit card payment.
// If PayPal,
// process the PayPal payment.
// If there is a payment problem, notify the user.
// Open a connection
// Set stored procedure parameters with the payment data.
// Call the save stored procedure.
```



```
    // -- Send an email receipt --
    // If the user requested a receipt
    // Get the customer data
    // Ensure a valid email address was provided.
    // If not,
    // request an email address from the user.
    // Open a connection
    // Set stored procedure parameters with the custo
    // Call the save stored procedure.
    // If a valid email address is provided,
    // Send an email.
}
```

Clear Purpose

Good Name

Focused Code

Short Length

Testable

Automated Code Test

Validation

Predictable Result

Exception Handling

Refactoring

- Restructuring code without changing its behavior
- Transform "code smells" into "clean code"

- Process:
  - Build unit tests
  - Refactor
  - Rerun tests

Clear Purpose

Good Name

Focused Code

Short Length

Testable

Automated Code Test

Validation

Predictable Result

Exception Handling

What's Wrong?

```
public decimal CalculatePercentOfGoalSteps(string goalSteps,
                                           string actualSteps)
{
  return Math.Round(decimal.Parse(actualSteps) /
                              decimal.Parse(goalSteps)*100M, 2);
}
```

- Clear Purpose
- Good Name
- Focused Code
- Short Length
- Testable
- Automated Code Test
- Validation
- Predictable Result
- Exception Handling

Guard Clauses

```
public decimal CalculatePercentOfGoalSteps(string goalSteps, string actualSteps)
{
   if (string.IsNullOrWhiteSpace(goalSteps))
      throw new ArgumentNullException(nameof(goalSteps));
   if (string.IsNullOrWhiteSpace(actualSteps))
      throw new ArgumentNullException(nameof(actualSteps));

   decimal goalStepCount = 0;
   if (!decimal.TryParse(goalSteps, out goalStepCount))
      throw new ArgumentException(nameof(goalSteps));

  decimal actualStepCount = 0;
  if (!decimal.TryParse(actualSteps, out actualStepCount))
      throw new ArgumentException(nameof(actualSteps));

  return Math.Round(actualStepCount / goalStepCount * 100M,2);
}
```

Method Overloading

```
private decimal CalculatePercentOfGoalSteps(decimal goalCount, decimal actualCount)
{
  if (goalCount <= 0) throw new ArgumentException(nameof(goalStepCount));
  return Math.Round(actualCount / goalCount * 100M,2);
}
```

```
public decimal CalculatePercentOfGoalSteps(string goalSteps, string actualSteps)
{
   if (string.IsNullOrWhiteSpace(goalSteps)) throw new ArgumentNullException(nameof(goalSteps));
   if (string.IsNullOrWhiteSpace(actualSteps)) throw new ArgumentNullException(nameof(actualSteps));

   decimal goalStepCount = 0;
   if (!decimal.TryParse(goalSteps, out goalStepCount)) throw new ArgumentException(nameof(goalSteps));

  decimal actualStepCount = 0;
  if (!decimal.TryParse(actualSteps, out actualStepCount)) throw new ArgumentException(nameof(actualSteps));

  return CalculatePercentOfGoalSteps(goalStepCount, actualStepCount);
}
```

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Returning Predictable Results

Value

Exceptions

Multiple Values

Null

Returning Multiple Values

- `ref` parameters

```
public decimal Calc(string goal, string actual, ref string message)
```

- `out` parameters

```
public decimal Calc(string goal, string actual, out string message)
```

- Tuples

```
public Tuple<decimal, string> Calc(string goal, string actual)
```

- object

```
public OperationResult<decimal> Calc(string goal, string actual)
```

Adding Exception Handling

Invalid User Entry

Invalid or Missing Data

Code Issues

System Issues

Use restrictive controls and binding

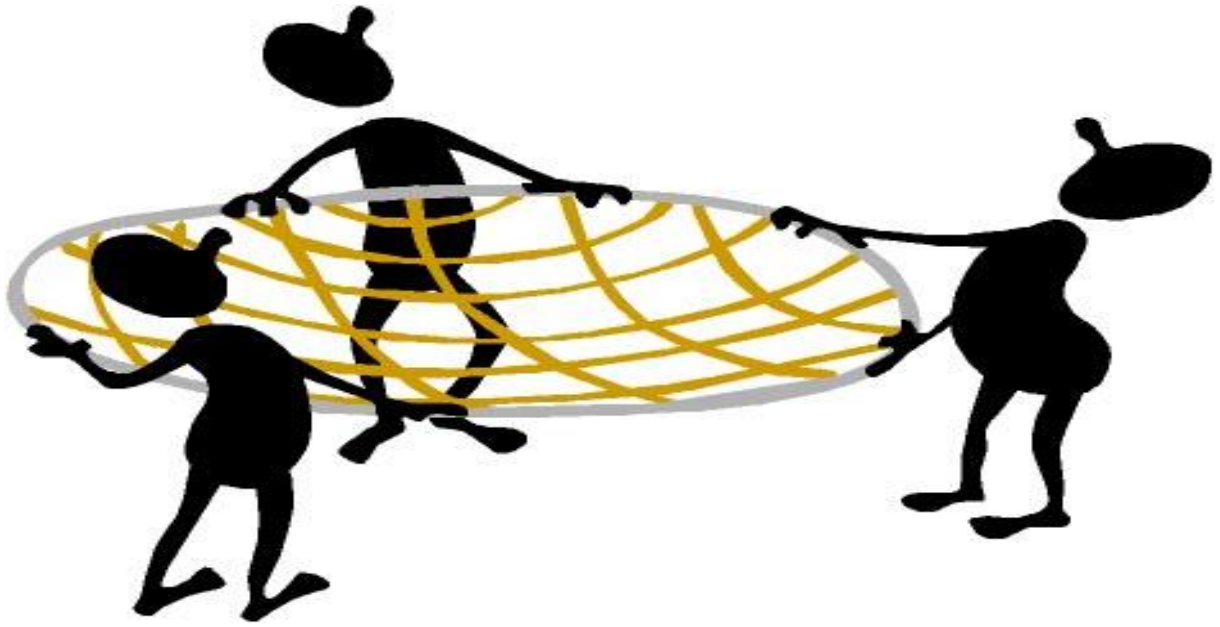
Use validation methods

Use good defaults

Return OperationResult

Throw exceptions

Global Exception Handler



Example

```
// For UI thread exceptions
Application.ThreadException +=
           new ThreadExceptionEventHandler(GlobalExceptionHandler);

// Force all Windows Forms errors to go through our handler.
Application.SetUnhandledExceptionMode(
                            UnhandledExceptionMode.CatchException);

// For non-UI thread exceptions
AppDomain.CurrentDomain.UnhandledException +=
       new UnhandledExceptionEventHandler(GlobalExceptionHandler);
```

Visual Studio LIVE!
EXPERT SOLUTIONS FOR .NET DEVELOPERS

Defensive Coding

Clean Code

- Improves Comprehension
- Simplifies Maintenance
- Reduces Bugs

Testable Code + Unit Tests

- Improves Quality
- Confirms Maintenance
- Reduces Bugs

Validation + Exception Handling

- Improves Predictability
- More Consistent
- Reduces Bugs

Thank You!

@deborahkurata

deborahk@insteptech.com

http://blogs.msmvps.com/deborahk

https://github.com/DeborahK/VSLive2015-NY