1 The request passed from the UI to the business process web service had two parameters that needed to be checked.

Code goal :

- If both where null or an empty string, then throw a business exception.
- If both where populated, throw a business exception.
- If only the Customer ID is populated, then check if the parameter maximum length isn't violated.
- If only the Product Number is populated, then check if the parameter maximum length isn't violated.
- Method return a boolean, but it was impossible to return a false value, so returning anything is redundant.

2

- Long Method
- Duplicated Code
- Comments
- Switch Statement

3 – See figures

1. The main `ValidateRequest` method

```
/// <span class="code-SummaryComment"><summary></span>
/// Validates the customer inquiry request.
/// <span class="code-SummaryComment"></summary></span>
/// <span class="code-SummaryComment"><param name="request">The customer inquiry request.
</param></span>
/// <span class="code-SummaryComment"><param name="customerFieldLength">Length of the
customer field.</param></span>
/// <span class="code-SummaryComment"><param name="productFieldLength">Length of the
product field.</param></span>
private void ValidateRequest( CustomerInquiryRequest request,
                              int customerFieldLength,
    int productFieldLength )
{
    // 1. Check both parameters are not null and not empty strings
    CheckCustomerInquiryNotNullOrEmpty( request );
    // 2. Check both parameters aren't populated
    CheckCustomerInquiryNullOrEmpty( request );
    // 3. Check CustomerID for field length and pad the parameter
    CheckCustomerIDValid( request, customerFieldLength );
    // 4. Check ProductNumber for field length and pad the parameter
    CheckProductNumberValid( request, productFieldLength );
}
```

2. Method to check both parameters are not null and not empty strings:

```
/// <span class="code-SummaryComment"><summary></span>
/// Checks the customer inquiry not null or empty.
/// <span class="code-SummaryComment"></summary></span>
/// <span class="code-SummaryComment"><param name="request"> customer inquiry request.
</param></span>
/// <span class="code-SummaryComment"><remarks>If both customer id and product number is
not null, then </span>
/// we throw a customer empty message business exception.<span class="code-
SummaryComment"></remarks></span>
private void CheckCustomerInquiryNotNullOrEmpty(
                CustomerInquiryRequest request )
{
    // Check both parameters are not null or empty string
    if( !string.IsNullOrEmpty( request.CustomerProduct.ProductNumber ) &&
        !string.IsNullOrEmpty( request.Customer.CustomerID ) )
    {
        // Both were populated
        throw new BusinessException(
              HandledErrors.InvalidBothParameterMessage );
    }
}
```

3. Method to check both parameters aren't populated:

```
/// <span class="code-SummaryComment"><summary></span>
/// Checks the customer inquiry null or empty.
/// <span class="code-SummaryComment"></summary></span>
/// <span class="code-SummaryComment"><param name="request">The customer inquiry request.
</param></span>
/// <span class="code-SummaryComment"><remarks></span>
/// If both customer id and product number is null, then we
/// throw a customer empty message business exception.<span class="code-SummaryComment">
</remarks></span>
private void CheckCustomerInquiryNullOrEmpty(
              CustomerInquiryRequest request )
{
    if( string.IsNullOrEmpty( request.Customer.CustomerID ) &&
        string.IsNullOrEmpty( request.CustomerProduct.ProductNumber ) )
    {
        // Both are null or empty string
        throw new BusinessException(
              HandledErrors.CustomerEmptyMessage );
    }
}
```

4. Method to check the `CustomerID` for field length and to pad the parameter to the correct length with zeroes:

```
/// <span class="code-SummaryComment"><summary></span>
/// Checks the customer ID valid.
/// <span class="code-SummaryComment"></summary></span>
/// <span class="code-SummaryComment"><param name="request">The customer inquiry request.
</param></span>
/// <span class="code-SummaryComment"><param name="customerIDFieldLength">Length of
</span>
///            the customer ID field.<span class="code-SummaryComment"></param></span>
/// <span class="code-SummaryComment"><remarks>Due to the Product number being null,
</span>
///        we are working with the customer id<span class="code-SummaryComment"></remarks>
</span>
private void CheckCustomerIDValid( CustomerInquiryRequest request,
                                   int customerIDFieldLength )
{
    if( string.IsNullOrEmpty( request.Customer.CustomerID ) )
    {
        // Check Customer ID length
        if( request.Customer.CustomerID.Length >
                        customerIDFieldLength )
        {
            throw new BusinessException(
                  HandledErrors.CustomerInvalidLengthMessage );
        }

        // Pad the left of the customer id
        request.Customer.CustomerID =
                request.Customer.CustomerID.PadLeft(
                customerIDFieldLength,
                Convert.ToChar( "0", CultureInfo.CurrentCulture ) );
    }
}
```

5. Method to check `ProductNumber` for field length and pad the parameter to the correct length with zeroes:

```
/// <span class="code-SummaryComment"><summary></span>
/// Checks the product number valid.
/// <span class="code-SummaryComment"></summary></span>
/// <span class="code-SummaryComment"><param name="request">The  customer inquiry request.
</param></span>
/// <span class="code-SummaryComment"><param name="productFieldLength">Length of </span>
///              the product field.<span class="code-SummaryComment"></param></span>
/// <span class="code-SummaryComment"><remarks>Due to the customer id being null,</span>
///    we are working with the product number<span class="code-SummaryComment"></remarks>
</span>
private void CheckProductNumberValid( CustomerInquiryRequest request,
                                      int productFieldLength )
{
    if( string.IsNullOrEmpty( request.CustomerProduct.ProductNumber ) )
    {
        // Check Product Length
        if( request.CustomerProduct.ProductNumber.Length >
                                      productFieldLength )
        {
            throw new BusinessException(
                  HandledErrors.ProductInvalidLengthMessage );
        }

        // Pad the left of the product number
        request.CustomerProduct.ProductNumber =
          request.CustomerProduct.ProductNumber.PadLeft(
          productFieldLength,
          Convert.ToChar( "0", CultureInfo.CurrentCulture ) );
    }
}
```