Comments

Cory House
PRINCIPAL CONSULTANT

@housecor www.cleancsharp.net

Agenda

Comments: A necessity and a crutch

Comments to avoid

Useful comments

Typical Interview

How do you write code that's readable and maintainable by others?
Uh...I use comments?

Comments lie.
They’re not kept up to date.
They’re either redundant
or a crutch for bad code.

I can’t understand your
lousy code ***without*** comments.
And…you smell like foot.

Are comments great, or a code smell?

Comments: A Necessity and a Crutch

Prefer expressive code over comments

Use comments when code isn’t sufficient

Comments to Avoid

- Redundant
- Intent
- Apology
- Warning
- Zombie Code
- Divider
- Brace Tracker
- Bloated Header
- Defect Log

Redundant Comments

🚫

```
int i = 1; // Set i = 1
```

🚫

```
var cory = new User(); //Instantiate a new user
```

🚫

```
/// <summary>
/// Default Constructor
/// </summary>
public User()
{
}
```

🚫

```
/// <summary>
/// Calculates Total Charges
/// </summary>
private void CalculateTotalCharges()
{
  //Total charges calculated here
}
```

Intent Comments

🚫

```
// Assure user’s account is deactivated.
if (user.Status == 2)
```

✓

```
if (user.Status == Status.Inactive)
```

Clarify intent without comments:

Improve function name

Declare intermediate variable

Declare a constant or enum

Extract conditional to function

Apology Comments

🚫 // Sorry, this crashes sometimes so I'm swallowing the exception.

🚫 // I was too tired to refactor this pile when I was done...

Don’t apologize

Fix it before commit/merge

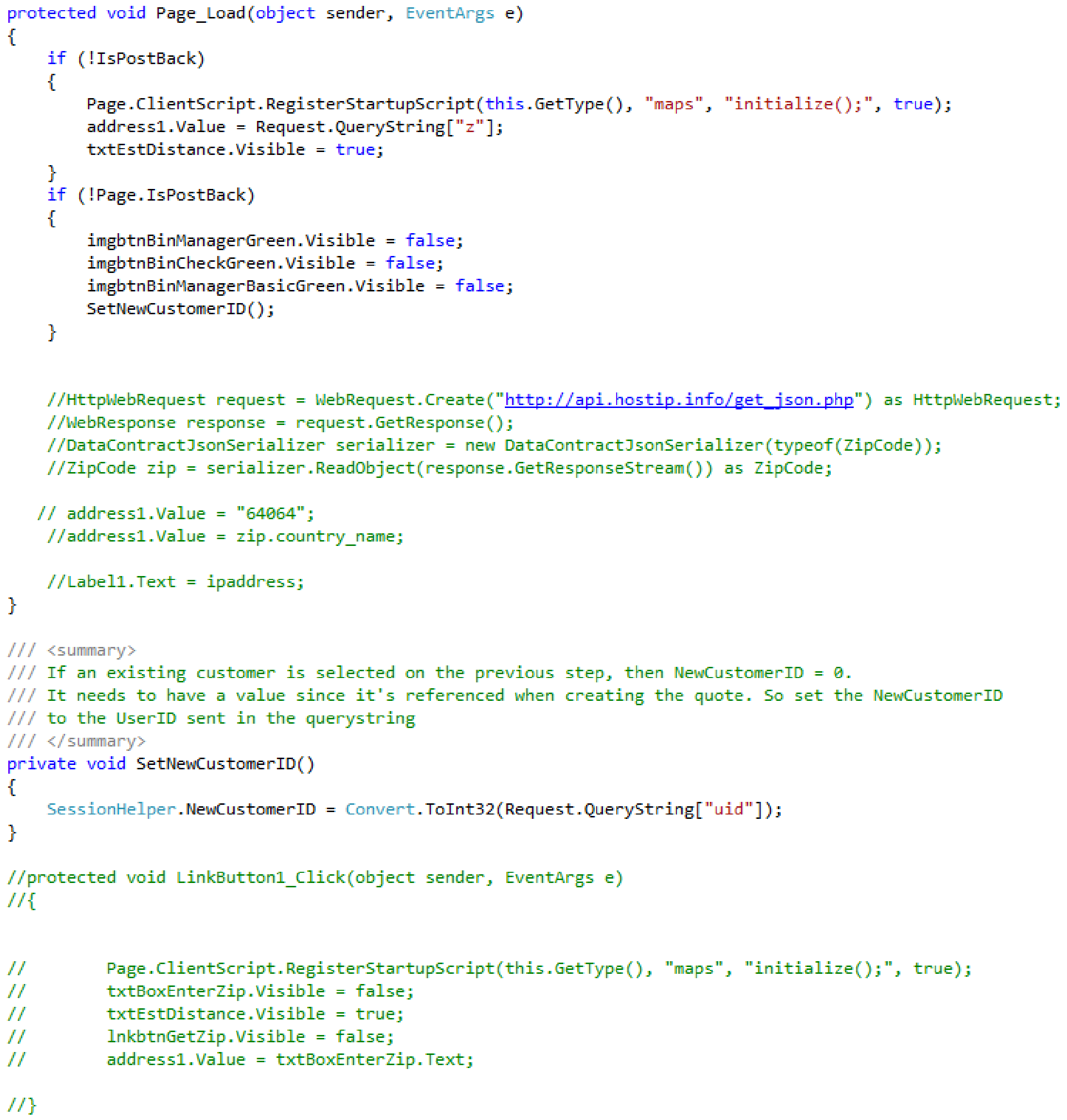
Add a TODO marker comment if you must

Warning Comments

🚫

```
// Here be dragons - see Bob

// Great sins against code below...
```

Kill Zombie Code



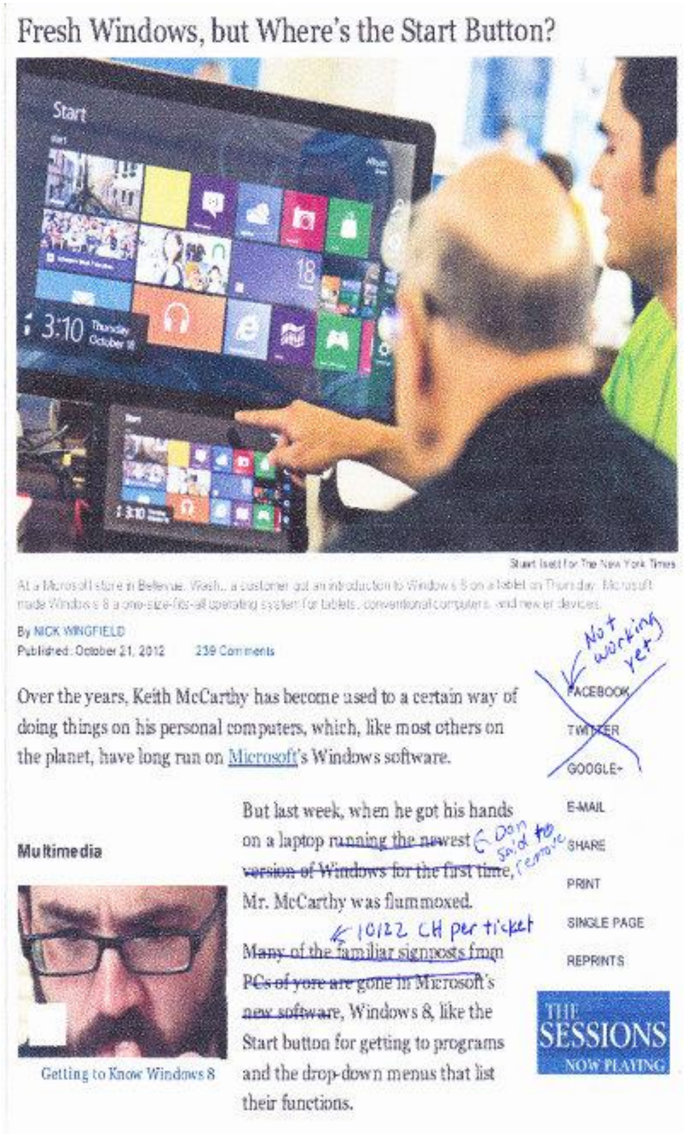
```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        Page.ClientScript.RegisterStartupScript(this.GetType(), "maps", "initialize();", true);
        address1.Value = Request.QueryString["z"];
        txtEstDistance.Visible = true;
    }
    if (!Page.IsPostBack)
    {
        imgbtnBinManagerGreen.Visible = false;
        imgbtnBinCheckGreen.Visible = false;
        imgbtnBinManagerBasicGreen.Visible = false;
        SetNewCustomerID();
    }


    //HttpWebRequest request = WebRequest.Create("http://api.hostip.info/get_json.php") as HttpWebRequest;
    //WebResponse response = request.GetResponse();
    //DataContractJsonSerializer serializer = new DataContractJsonSerializer(typeof(ZipCode));
    //ZipCode zip = serializer.ReadObject(response.GetResponseStream()) as ZipCode;

   // address1.Value = "64064";
    //address1.Value = zip.country_name;

    //Label1.Text = ipaddress;
}

/// <summary>
/// If an existing customer is selected on the previous step, then NewCustomerID = 0.
/// It needs to have a value since it's referenced when creating the quote. So set the NewCustomerID
/// to the UserID sent in the querystring
/// </summary>
private void SetNewCustomerID()
{
    SessionHelper.NewCustomerID = Convert.ToInt32(Request.QueryString["uid"]);
}

//protected void LinkButton1_Click(object sender, EventArgs e)
//{


//        Page.ClientScript.RegisterStartupScript(this.GetType(), "maps", "initialize();", true);
//        txtBoxEnterZip.Visible = false;
//        txtEstDistance.Visible = true;
//        lnkbtnGetZip.Visible = false;
//        address1.Value = txtBoxEnterZip.Text;

//}
```

Common Causes

Risk Aversion

Hoarding mentality

Optimize the Signal to Noise Ratio



Fresh Windows, but Where's the Start Button?

Stuart Isett for The New York Times

At a Microsoft store in Bellevue, Wash., a customer got an introduction to Windows 8 on a tablet on Thursday. Microsoft made Windows 8 a one-size-fits-all operating system for tablets, conventional computers, and newer devices.

By NICK WINGFIELD
Published: October 21, 2012   239 Comments

Over the years, Keith McCarthy has become used to a certain way of doing things on his personal computers, which, like most others on the planet, have long run on Microsoft's Windows software.

But last week, when he got his hands on a laptop ~~running the newest version of Windows for the first time~~, ← Don said to remove Mr. McCarthy was flummoxed.

← 10/22 CH per ticket

~~Many of the familiar signposts from PCs of yore are gone in Microsoft's new software~~, Windows 8, like the Start button for getting to programs and the drop-down menus that list their functions.

Multimedia

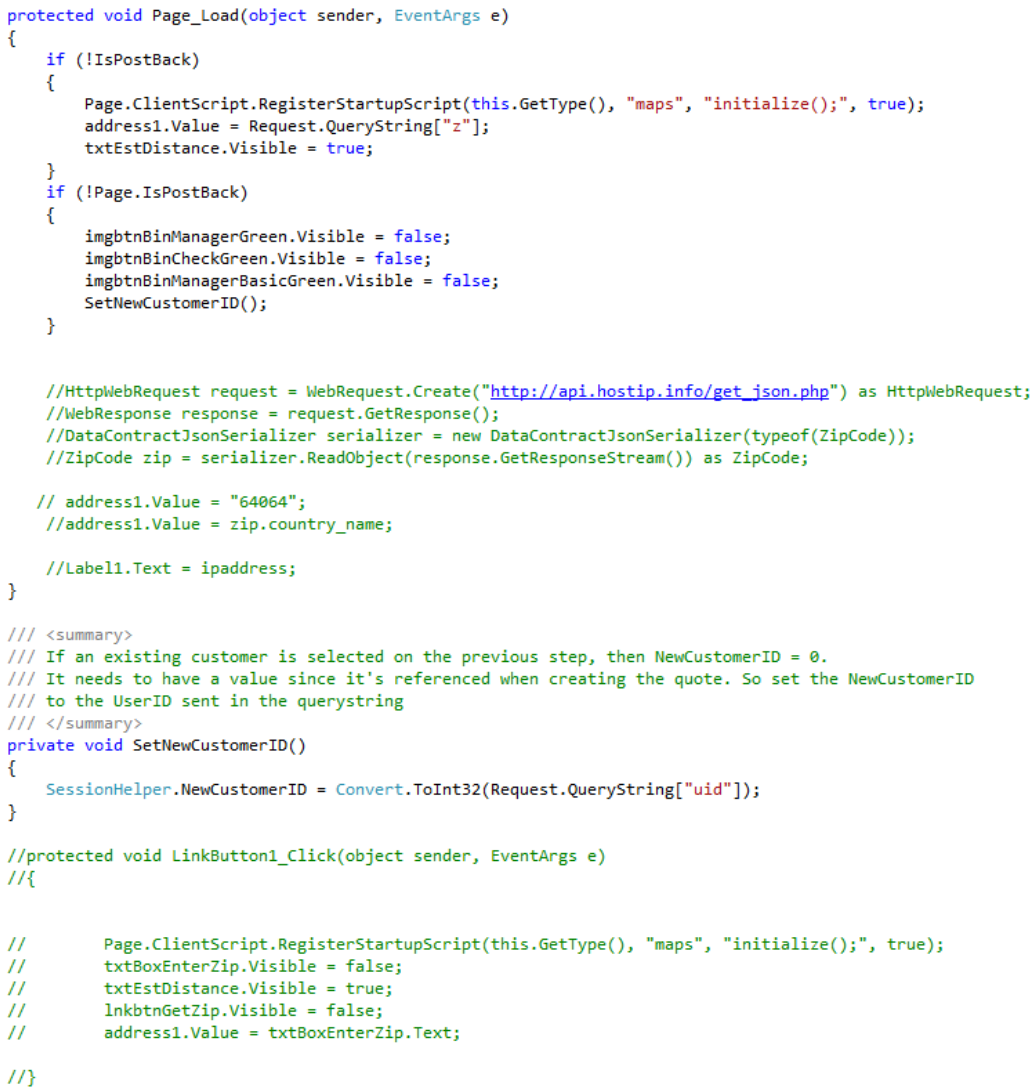
Getting to Know Windows 8

Not working yet

~~FACEBOOK~~
~~TWITTER~~
~~GOOGLE+~~
E-MAIL
SHARE
PRINT
SINGLE PAGE
REPRINTS

We wouldn't stand for this.

Ambiguity Hinders Debugging



```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        Page.ClientScript.RegisterStartupScript(this.GetType(), "maps", "initialize();", true);
        address1.Value = Request.QueryString["z"];
        txtEstDistance.Visible = true;
    }
    if (!Page.IsPostBack)
    {
        imgbtnBinManagerGreen.Visible = false;
        imgbtnBinCheckGreen.Visible = false;
        imgbtnBinManagerBasicGreen.Visible = false;
        SetNewCustomerID();
    }


    //HttpWebRequest request = WebRequest.Create("http://api.hostip.info/get_json.php") as HttpWebRequest;
    //WebResponse response = request.GetResponse();
    //DataContractJsonSerializer serializer = new DataContractJsonSerializer(typeof(ZipCode));
    //ZipCode zip = serializer.ReadObject(response.GetResponseStream()) as ZipCode;

   // address1.Value = "64064";
    //address1.Value = zip.country_name;

    //Label1.Text = ipaddress;
}

/// <summary>
/// If an existing customer is selected on the previous step, then NewCustomerID = 0.
/// It needs to have a value since it's referenced when creating the quote. So set the NewCustomerID
/// to the UserID sent in the querystring
/// </summary>
private void SetNewCustomerID()
{
    SessionHelper.NewCustomerID = Convert.ToInt32(Request.QueryString["uid"]);
}

//protected void LinkButton1_Click(object sender, EventArgs e)
//{


//        Page.ClientScript.RegisterStartupScript(this.GetType(), "maps", "initialize();", true);
//        txtBoxEnterZip.Visible = false;
//        txtEstDistance.Visible = true;
//        lnkbtnGetZip.Visible = false;
//        address1.Value = txtBoxEnterZip.Text;

//}
```

What did this section do?
Was this accidentally commented?
Who did this?
Do I need to refactor this too?
How does my change impact this code?
What if someone uncomments it later?

Kill Zombie Code

- Reduces readability
- Creates ambiguity
- Hinders refactoring
- Adds noise to searches
- Code isn’t “lost” anyway

Kill Zombie Code Checklist

About to comment out code? Ask:

When would this be uncommented?

Can I just get it from source control later?

Is this incomplete work that should be worked via a branch?

Should this be toggled via configuration?

Did I refactor out the need for this code?

Divider Comments

```
private void MyLongFunction()
{
  // lots
  // of
  // code

  // Start search for available concert tickets

  // lots
  // of
  // concert
  // search
  // code

  // End of concert ticket search

  // lots
  // more
  // code
}
```

Brace Tracker Comments

🚫

```
private void AuthenticateUsers()
{
 bool validLogin = false;
 //deeply
  //nested
   //code
   if (validLogin)
   {
    // lots
    // of
    // code
    // here
   } // end user login
  //even
 //more code
}
```

✅

```
private void AuthenticateUsers()
{
 bool validLogin = false;
 //deeply
  //nested
   //code
   if (validLogin)
   {
    LoginUser();
   }
  //even
 //more code
}
```

Avoid line endings

Don’t repeat yourself

Follow conventions

Bloated Header

🚫

```
//***************************************************
// Filename: Monolith.cs            *
//                    *
// Author: Cory House             *
// Created: 12/20/2019            *
// Weather that day: Patchy fog, then snow    *
//                    *
// Summary                 *
// This class does a great many things. To make it *
// extra useful I placed pretty much all the app  *
// logic here. You wish your class was this    *
// powerful. Bwahhahha!            *
//***************************************************
```

Defect Log

```
// Defect #5274 DA 12/10/2010
// We weren't checking for null here
if (firstName != null)
{
  //...
}
```

Change metadata belongs in source control

A well written book isn’t covered in author notes

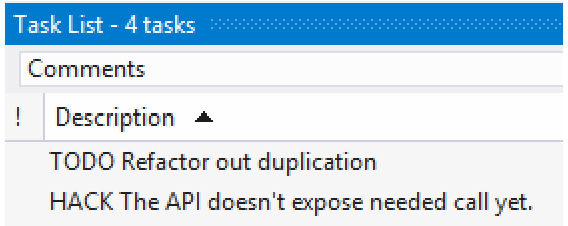
Clean Comments

- To Do
- Summary
- Documentation

To Do Comments

✓

```
// TODO Refactor out duplication

// HACK The API doesn't expose needed call yet.
```



Task List - 4 tasks

Comments

! Description ▲

TODO Refactor out duplication

HACK The API doesn't expose needed call yet.

Standardize

Watch Out

- May be an apology or warning comment in disguise
- Often ignored

Summary Comments

✓

```
// Encapsulates logic for calculating retiree benefits

// Generates custom newsletter emails
```

Describes intent at general level higher than the code

Often useful to provide high level overview of classes

Risk: Don’t use to simply augment poor naming/code level intent

Documentation

✓ `// See microsoft.com/api for documentation`

Useful when it can’t be expressed in code.

About to Write a Comment?

Comments are useful, but generally a last resort.

Ask:

1. **Could I express what I’m about to type in *code*?**
    - Intermediate variable, eliminate magic number, utilize enum?
    - Refactor to a well-named method.
        - Separate scope
        - More likely to stay updated
        - Better testability

2. **Am I explaining bad code I just wrote instead of refactoring?**

3. **Should this be a source control commit message?**

Summary

Comments: A necessity and a crutch

Goal: Self-documenting code

Try to convey intent without comments

Avoid

- Redundant, Intent, Apology, Warning, Zombie Code, Divider, Brace Tracker, Bloated Header, Defect Log

Clean

- To do, summary, documentation

Next: Refactoring to Clean C#