Clean Code: Writing Code for Humans

Functions

Cory House
BitNative.com
Twitter: @housecor

pluralsight
hardcore developer training

Agenda

- High signal functions
- When to create a function
- Techniques to maintain simplicity
- Code smells and refactoring techniques
- Error handling

Function vs Method

- Both functions and methods are pieces of code, called by name.
- Core difference: Methods are associated with an object.
- In this module: functions and methods are the same thing.

When to create a function

Duplication

Indentation

Unclear intent

> 1task

1) Duplication

Key: Don’t repeat yourself

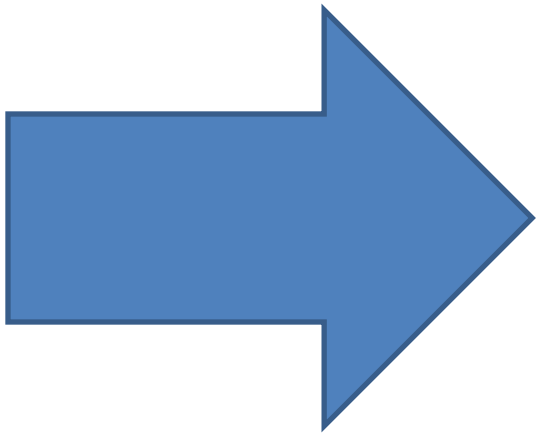
Code is a liability

Less is more

Look for Patterns

```
if (!string.IsNullOrEmpty(ws.SEOTargetLocation1) && ws.SEOTargetLocation1.Contains(","))
{
        string[] pieces = ws.SEOTargetLocation1.Split(",".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);
        if (pieces.Length == 2 && pieces[1].Trim().Length == 2)
        {
                string dl1_url = BuildDealerUrl(auto.Make, pieces[0], pieces[1]);
                string dl1_text = string.Format("<a href=\"{0}\">{1} {2} {4}, {5}</a>", dl1_url, auto.YearName ?? 0, auto.Make, auto.Model, pieces[0], pieces[1]);

                _DisclaimerUrls.Text += dl1_text + " ";
        }
}

if (!string.IsNullOrEmpty(ws.SEOTargetLocation2) && ws.SEOTargetLocation2.Contains(","))
{
        string[] pieces = ws.SEOTargetLocation2.Split(",".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);
        if (pieces.Length == 2 && pieces[1].Trim().Length == 2)
        {
                string dl1_url = BuildDealerUrl(auto.Make, pieces[0], pieces[1]);
                string dl1_text = string.Format("<a href=\"{0}\">{1} {2} {4}, {5}</a>", dl1_url, auto.YearName ?? 0, auto.Make, auto.Model, pieces[0], pieces[1]);

                _DisclaimerUrls.Text += dl1_text + " ";
        }
}

if (!string.IsNullOrEmpty(ws.SEOTargetLocation3) && ws.SEOTargetLocation3.Contains(","))
{
        string[] pieces = ws.SEOTargetLocation3.Split(",".ToCharArray(), StringSplitOptions.RemoveEmptyEntries);
        if (pieces.Length == 2 && pieces[1].Trim().Length == 2)
        {
                string dl1_url = BuildDealerUrl(auto.Make, pieces[0], pieces[1]);
                string dl1_text = string.Format("<a href=\"{0}\">{1} {2} {4}, {5}</a>", dl1_url, auto.YearName ?? 0, auto.Make, auto.Model, pieces[0], pieces[1]);

                _DisclaimerUrls.Text += dl1_text + " ";
        }
}
```

2) Excessive Indentation: Arrow Code



```
if
    if
        if
            if
                do stuff
            endif
        endif
    endif
endif
```

Comprehension decreases beyond three levels of nested 'if' blocks.

Excessive Indentation: Solutions

- Extract Method
- Fail Fast
- Return Early

Extract Method

Before

```
if
    if
        while
            do
            some
            complicated
            thing
        end while
    end if
end if
```

After

```
if
    if
        doComplicatedThing()
    end if
end if
```

```
doComplicatedThing()
{
    while
        do some complicated thing
    end while
}
```

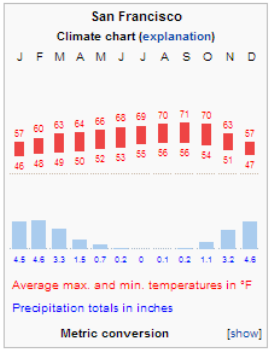
Extracting a method: like footnotes

Climate [edit source | edit beta]

A popular quote incorrectly attributed to Mark Twain is "The coldest winter I ever spent was a summer in San Francisco".[68][69] San Francisco's climate is characteristic of the cool-summer Mediterranean climate (Csb)[70] of California's coast, "generally characterized by moist mild winters and dry summers".[71] Since it is surrounded on three sides by water, San Francisco's weather is strongly influenced by the cool currents of the Pacific Ocean, which moderate temperature swings and produce a remarkably mild year-round climate with little seasonal temperature variation.

Fog is a regular feature of San Francisco summers.

Among major U.S. cities, San Francisco has the coldest daily mean, maximum, and minimum temperatures for June, July, and August.[72] During the summer, rising hot air in California's interior valleys creates a low pressure area that draws winds from the North Pacific High through the Golden Gate, which creates the city's characteristic cool winds and fog.[73] The fog is less pronounced in eastern neighborhoods and during the late summer and early fall, which is the warmest time of the year.



Because of its sharp topography and maritime influences, San Francisco exhibits a multitude of distinct microclimates. The high hills in the geographic center of the city are responsible for a 20% variance in annual rainfall between different parts of the city. They also protect neighborhoods directly to their east from the foggy and sometimes very cold and windy conditions experienced in the Sunset District; for those who live on the eastern side of the city, San Francisco is sunnier, with an average of 260 clear days, and only 105 cloudy days per year.

Temperatures exceed 75 °F (24 °C) on average only 29 days a year.[74] The dry period of May to October is mild to warm, with average high temperatures of 64–71 °F (18–22 °C) and lows of 51–56 °F (11–13 °C). The rainy period of November to April is slightly cooler, with high temperatures of 58–64 °F (14–18 °C) and lows of 46–51 °F (8–11 °C). On average, there are 73 rainy days a year, and annual precipitation averages 23.6 inches (599.44 mm). Snowfall in the city is very rare, with only 10 measurable accumulations recorded since 1852, most recently in 1976 when up to 5 inches (130 mm) fell on Twin Peaks.[75][76]

The highest recorded temperature at the official National Weather Service office was 103 °F (39 °C) on July 17, 1988, and June 14, 2000. The lowest recorded temperature was 27 °F (−3 °C) on December 11, 1932.[77] The National Weather Service provides a helpful visual aid[78] graphing the information in the table below to display visually by month the annual typical temperatures, the past year's temperatures, and record temperatures.

Climate data for San Francisco (downtown), 1981–2010 normals [hide]

Month	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Year
Record high °F (°C)	79 (26)	81 (27)	86 (30)	94 (34)	101 (38)	103 (39)	103 (39)	98 (37)	101 (38)	102 (39)	86 (30)	76 (24)	103 (39)
Average high °F (°C)	56.9 (13.8)	60.2 (15.7)	61.8 (16.6)	63.1 (17.3)	64.3 (17.9)	66.4 (19.1)	66.5 (19.2)	68.1 (20.1)	70.2 (21.2)	69.2 (20.7)	63.1 (17.3)	57.1 (13.9)	63.9 (17.7)
Average low °F (°C)	45.7 (7.6)	47.5 (8.6)	48.5 (9.2)	49.2 (9.6)	51.0 (10.6)	52.8 (11.6)	54.1 (12.3)	55.1 (12.8)	55.1 (12.8)	53.7 (12.1)	50.1 (10.1)	46.1 (7.8)	50.7 (10.4)
Record low °F (°C)	29 (−2)	31 (−1)	33 (1)	40 (4)	42 (6)	46 (8)	47 (8)	46 (8)	47 (8)	43 (6)	38 (3)	27 (−3)	27 (−3)
Rainfall inches (mm)	4.50 (114.3)	4.45 (113)	3.25 (82.6)	1.46 (37.1)	0.70 (17.8)	0.16 (4.1)	0.00 (0)	0.06 (1.5)	0.21 (5.3)	1.12 (28.4)	3.16 (80.3)	4.56 (115.8)	23.63 (600.2)
Avg. rainy days (≥ 0.01 in)	11.7	11.1	11.0	6.5	3.8	1.5	0.3	1.0	1.7	3.9	8.9	11.6	73.0
Mean monthly sunshine hours	185.9	207.7	269.1	309.3	325.1	311.4	313.3	287.4	271.4	247.1	173.4	160.6	3,061.7

Source: NOAA (extremes 1874–present, sun 1961–1974)[79][80]

74. ^ "Weatherbase: Historical Weather for San Francisco, California. Summary of weather data." . weatherbase.com.
75. ^ Climate of San Francisco: Snowfall Golden Gate Weather Services. Retrieved December 3, 2006.

original on July 26, 2010. Retrieved July 26, 2010.

211. ^ Wallace Ravven (July 22, 2003). "New UCSF Mission Bay campus: country's largest biomedical university expansion" . UCSF. Retrieved June 9, 2008.

Return Early

```
private bool ValidUsername(string username)
{
    bool isValid = false;

    const int MinUsernameLength = 6;
    if (username.Length >= MinUsernameLength)
    {
        const int MaxUsernameLength = 25;
        if (username.Length <= MaxUsernameLength)
        {
            bool isAlphaNumeric = username.All(Char.IsLetterOrDigit);
            if (isAlphaNumeric)
            {
                if (!ContainsCurseWords(username))
                {
                    isValid = IsUniqueUsername(username);
                }
            }
        }
    }
    return isValid;
}
```

Return Early

```
private bool ValidUsername(string username)
{
    const int MinUsernameLength = 6;
    if (username.Length < MinUsernameLength) return false;

    const int MaxUsernameLength = 25;
    if (username.Length > MaxUsernameLength) return false;

    bool isAlphaNumeric = username.All(Char.IsLetterOrDigit);
    if (!isAlphaNumeric) return false;

    if (ContainsCurseWords(username)) return false;

    return IsUniqueUsername(username);
}
```

Return Early

Use a return when it enhances readability... In certain routines, once you know the answer…not returning immediately means that you have to write more code.

Steve McConnell, "Code Complete"

We Return Early in Real Life

```
private bool ValidUsername(string username)
{
    const int MinUsernameLength = 6;
    if (username.Length < MinUsernameLength) return false;

    const int MaxUsernameLength = 25;
    if (username.Length > MaxUsernameLength) return false;

    bool isAlphaNumeric = username.All(Char.IsLetterOrDigit);
    if (!isAlphaNumeric) return false;

    if (ContainsCurseWords(username)) return false;

    return IsUniqueUsername(username);
}
```

Avoid Arrow Code: Fail Fast

Dirty

```
public void RegisterUser(string username, string password)
{
    if (!string.IsNullOrWhiteSpace(username))
    {
        if (!string.IsNullOrWhiteSpace(password))
        {
            //register user here.
        }
        else
        {
            throw new ArgumentException("Username is required.");
        }
    }
    else
    {
        throw new ArgumentException("Password is required");
    }
}
```

Clean

```
public void RegisterUser(string username, string password)
{
    if (string.IsNullOrWhiteSpace(username)) throw new ArgumentException("Username is required.");
    if (string.IsNullOrWhiteSpace(password)) throw new ArgumentException("Password is required");

    //register user here.
}
```

Guard Clauses

Fail Fast

```
private void LoginUser(User user)
{
    switch (user.Status)
    {
        case Status.Active:
            //logic for active users
            break;
        case Status.Inactive:
            //logic for inactive users
            break;
        case Status.Locked:
            //logic for locked users
            break;
        default:
            throw new ApplicationException("Unknown user status: " + user.Status);
    }
}
```

3) Convey Intent

Dirty

```
//Check for valid file extensions. Confirm admin or active
if (fileExtension == "mp4" ||
    fileExtension == "mpg" ||
    fileExtension == "avi")
    && (isAdmin || isActiveFile);
```

Clean

```
if (ValidFileRequest(fileExtension, active))


private bool ValidFileRequest(string fileExtension, bool isActiveFile, bool isAdmin)
{
    var validFileExtensions = new List<string>() { "mp4", "mpg", "avi" };

    bool validFileType = validFileExtensions.Contains(fileExtension);
    bool userIsAllowedToViewFile = isActiveFile || isAdmin;

    return validFileType && userIsAllowedToViewFile;
}
```

4) Do one thing

Aids the reader

Promotes reuse

Eases naming and testing

Avoids side-effects

Could you read a book with no paragraphs?

Mayfly variables

Would you read this?
Bobby - Boy from Chicago
Sandy – Girl from New York
Tom – Man from Boston
Etc...

Once upon a time…

Without context, it's noise and mental

Mayfly Variables

Only live a few hours!

Recipe for Mayfly variables

1. Initalize Variables just-in-time
2. Do one thing

```
private void Mayfly()
{
    bool a = false;
    int b = 0;
    string c = string.Empty;
    bool d = true;

    //body continues
    //...
    //...
    //...
    //...
    //...
    //...

    a = SomethingIsTrue();

    if (a)
    {
        if (c.Length > b)
        {
            //body continues
            //...
            //...
            //...
            //...
            //...
            //...

            d = c.Substring(0, 3) == b.ToString();
        }
    }
}
```

How many parameters?

- Strive for 0 - 2 parameters
- Easier to understand
- Easier to test
- Helps assure function does one thing

Dirty

```
public void SaveUser(User user, bool sendEmail, int emailFormat,
    bool printReport, bool sendBill)
```

Clean

```
private void SaveUser(User user)
```

Watch for Flag Arguments

- A sign the function is doing two things.

Dirty

```
private void SaveUser(User user, bool emailUser)
{
    //save user

    if (emailUser)
    {
        //email user
    }
}
```

Clean

```
private void SaveUser(User user)
{
    //save user
}

private void EmailUser(User user)
{
    //email user
}
```

Signs it's too long?

Rarely be over 20 lines
Hardly ever over 100 lines
No more than 3 parameters
Robert C. Martin , "Clean Code"

Bottom Line

The maximum length…is inversely proportional to the complexity and indentation level of that function. So, if you have a conceptually simple function that is just one long (but simple) case statement…it's OK to have a longer function…if you have a complex function…adhere to limits all the more closely.

Linux style guide

Simple functions can be longer. Complex functions should be short.

Kinds of Exceptions

Unrecoverable	Recoverable	Ignorable
• Null reference • File not found • Access denied	• Retry connection • Try different file • Wait and try again	• Logging click

Try/Catch/Log = Fail Slow

Dirty

```
try
{
    RegisterSpeaker();
}
catch(Exception e)
{
    LogError(e);
}

EmailSpeaker();
```

Clean

```
RegisterSpeaker();
EmailSpeaker();
```

Try/Catch Body Standalone

Dirty

```
try
{
    //many
    //lines
    //of
    //complicated
    //and
    //verbose
    //logic
    //here
}
catch (ArgumentOutOfRangeException)
{
    //do something here
}
```

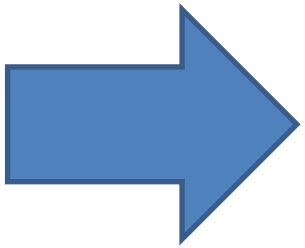
Clean

```
try
{
    SaveThePlanet();
}
catch (ArgumentOutOfRangeException)
{
    //do something here
}


private void SaveThePlanet()
{
        //many
        //lines
        //of
        //complicated
        //and
        //verbose
        //logic
        //here
}
```

Summary

When to create a function

- Duplication
- Indentation
- Unclear intent
- > 1task



Signs its too long?

Kinds of Exceptions

Unrecoverable	Recoverable	Ignorable
• Null reference • File not found • Access denied	• Retry connection • Try different file • Wait and try again	• Logging click