Writing Clean Methods

Cory House
PRINCIPAL CONSULTANT

@housecor   www.cleancsharp.net

Agenda

High signal functions

When to create a function

Techniques to maintain simplicity

Approaches to minimize variable lifetime

Signs a function is too long

Exceptions and 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, I use the terms interchangeably.

When To Create a Function

Duplication

Indentation

Unclear intent

> 1 task

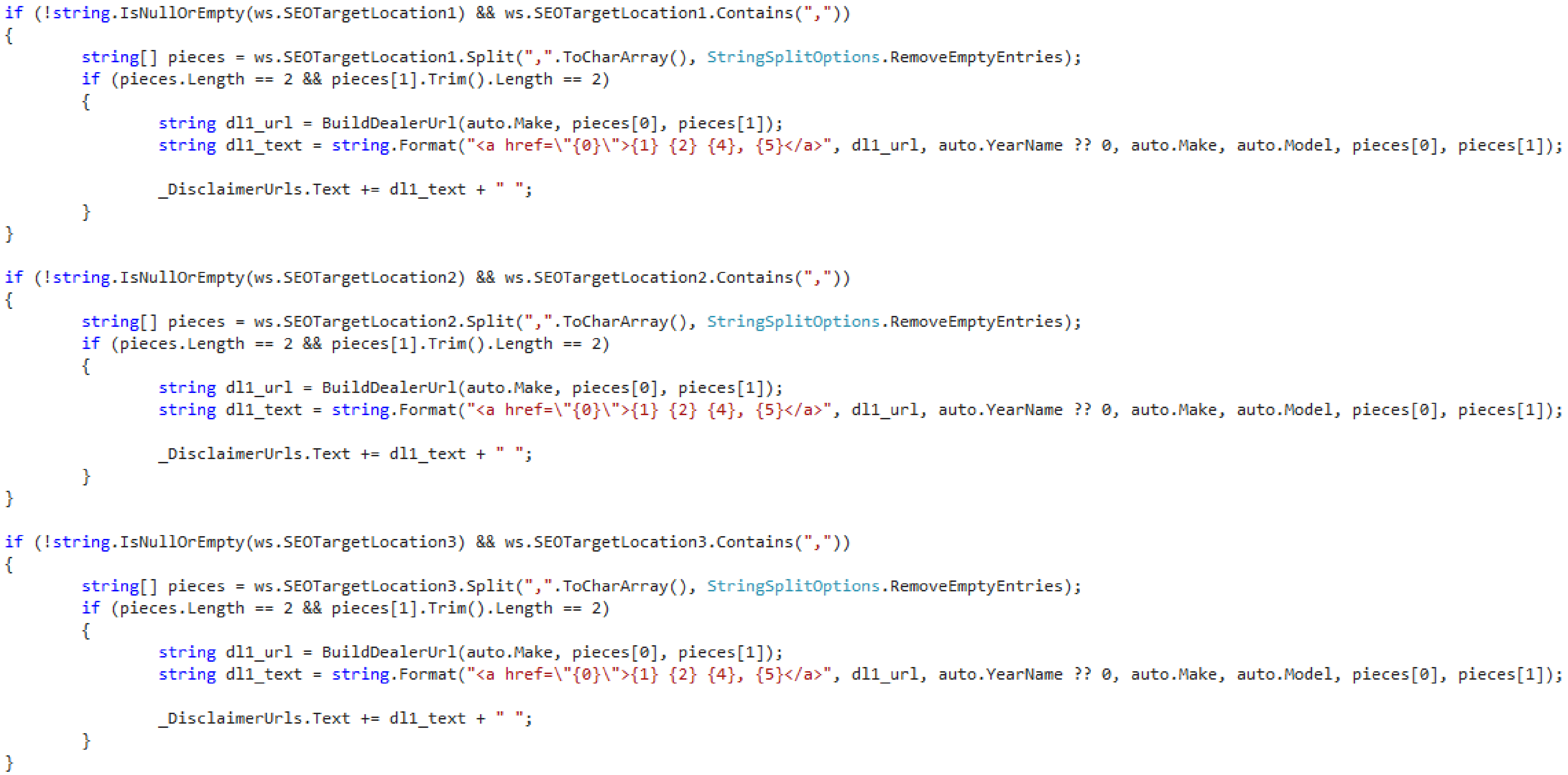
Duplication

Key: Don’t repeat yourself

Code is a liability

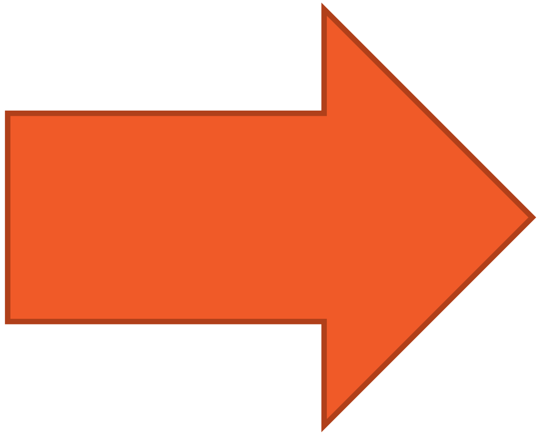
Less is more

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 + " ";
        }
}
```

Excessive Indentation: Arrow Code



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

Comprehension decreases beyond three levels of nested 'if' blocks.
Study by Noam Chomsky and Gerald Weinberg

Excessive Indentation: Solutions

Extract Method

Fail Fast

Return Early

Extract Method

Before

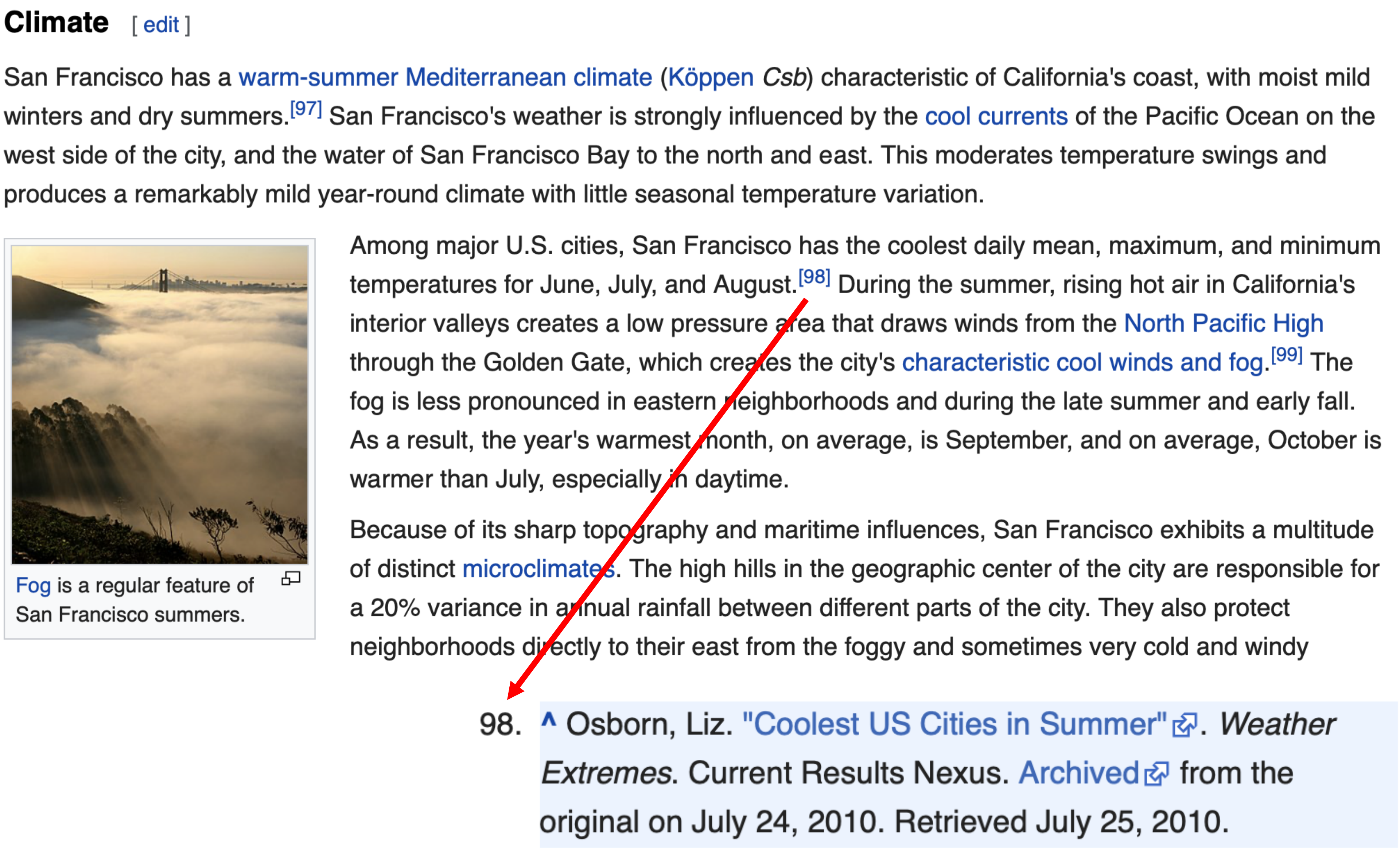
```
if
  if
   while
    do
    some
    complicated
    thing
   end while
  end if
 end if
```

After

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

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

Extracting a Method: Like Footnotes



Climate [edit]

San Francisco has a warm-summer Mediterranean climate (Köppen *Csb*) characteristic of California's coast, with moist mild winters and dry summers.[97] San Francisco's weather is strongly influenced by the cool currents of the Pacific Ocean on the west side of the city, and the water of San Francisco Bay to the north and east. This moderates temperature swings and produces 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 coolest daily mean, maximum, and minimum temperatures for June, July, and August.[98] 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.[99] The fog is less pronounced in eastern neighborhoods and during the late summer and early fall. As a result, the year's warmest month, on average, is September, and on average, October is warmer than July, especially in daytime.

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

98. ^ Osborn, Liz. "Coolest US Cities in Summer". *Weather Extremes*. Current Results Nexus. Archived from the original on July 24, 2010. Retrieved July 25, 2010.

Fail Fast

⊘

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

✓

```
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
}
```

Guard Clauses

Fail Fast

✓

```
public 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 status: " + user.Status);
 }
}
```

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);
}
```

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);
}
```

Convey Intent

🚫

```
 //Check for valid file extensions, confirm is admin or active
 if ( (fileExt == ".mp4" || fileExt == ".mpg" || fileExt == ".avi") &&
     (isAdmin == 1 || isActiveFile))
```

✅

```
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;
}
```

Do One Thing

Aids the reader

Promotes reuse

Eases naming and testing

Avoids side-effects

Would you read a book without 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 weight

Mayfly Variables

```
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();
    }
  }
}
```

Only live a few hours!

Mayfly variable recipe

1. Initialize variables just-in-time
2. Do one thing

How Many Parameters?

🚫

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

✓

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

Avoid Flag Arguments

🚫

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

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

✅

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

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

Signs It’s Too Long

Whitespace & Comments

Scrolling required

Naming issues

Multiple Conditionals

Hard to digest

Rarely be over 20 lines
Rarely over 3 parameters

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.

Exception Types

Unrecoverable

- Null reference
- File not found
- Access denied

Recoverable

- Retry connection
- Try different file
- Wait and try again

Ignorable

- Logging click

Try/Catch/Log = Fail Slow

🚫

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

EmailSpeaker();
```

✅

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

Try/Catch Body Standalone

🚫

```
try
{
  // many lines
  // of complicated
  // logic here
}
  catch(Exception e)
{
  // do something here
}
```

✅

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

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

Summary

When to create a function

- Duplication, indentation, unclear intent, multiple tasks

Eliminate indentation

- Extract method, fail fast, return early

Limit variable lifetime (Mayfly variables)

Avoid boolean args

Watch for signs the func is too long

Avoid catching unrecoverable exceptions

Next up: Classes