

# Code Quality



Somkiat Puisungnoen

Search

Somkiat | Home

Update Info 1 View Activity Log 10+ ...

Timeline About Friends 3,138 Photos More

When did you work at Opendream? X

... 22 Pending Items

Post Photo/Video Live Video Life Event

What's on your mind?

Public Post

Intro

Software Craftsmanship

Software Practitioner at สยามชานนาภิเษก พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Somkiat Puisungnoen 15 mins · Bangkok · ...

Java and Bigdata



Page

Messages

Notifications 3

Insights

Publishing Tools

Settings

Help ▾



somkiat.cc

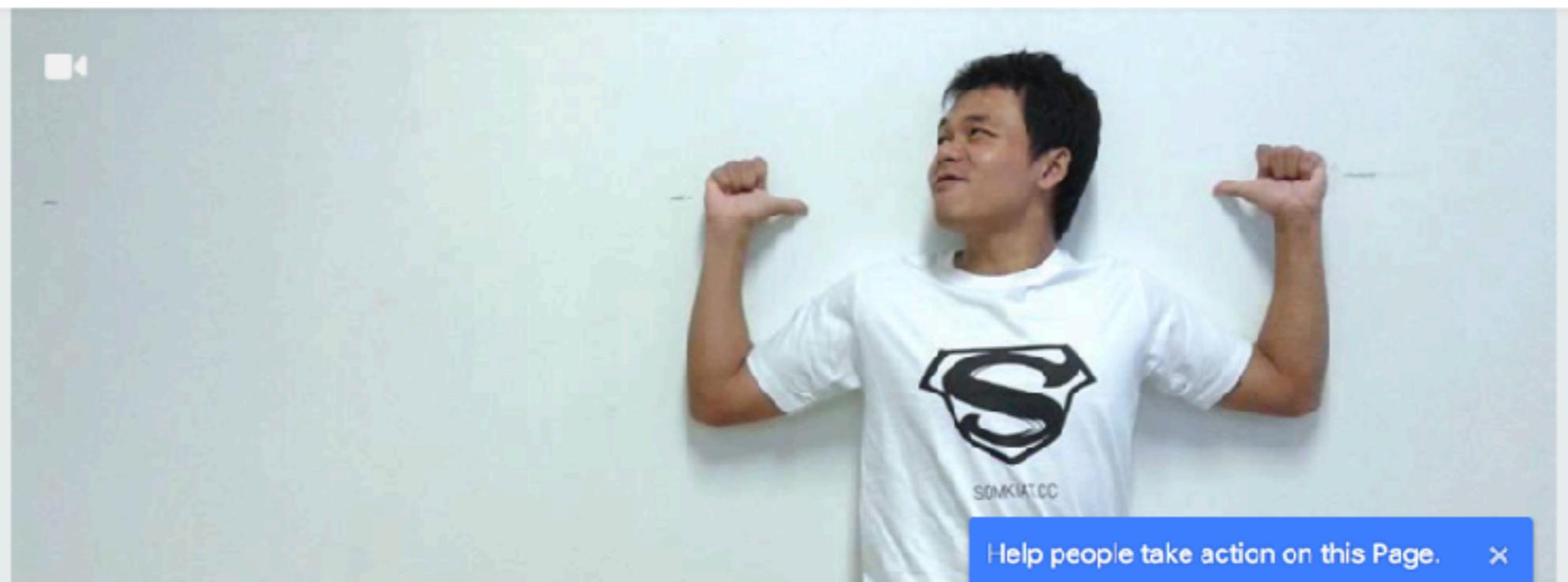
@somkiat.cc

Home

Posts

Videos

Photos



 Liked ▾

 Following ▾

 Share

...

+ Add a Button



Code Quality

3

© 2017 - 2018 Siam Chamnankit Company Limited. All rights reserved.

# Code Quality



# Code qualities

Efficiency  
Performance  
Integrity  
Recoverability  
Simplicity  
Maintainability  
Flexibility  
Valuable (customer/user)



# Code qualities

Efficiency  
**Performance**  
Integrity  
Recoverability  
**Simplicity**  
**Maintainability**  
Flexibility  
Valuable (customer/user)



# Coding is for Human



Programming is the art of telling  
another **human** what one wants  
the computer to do.

*- Donald Knuth -*



**Any fool** can write code that a  
computer can understand.

**Good** programmers write code  
that humans can understand.

*- Martin Fowler -*



# Why should I care ?

Reading is hard

Justification

Technical debt is depressing

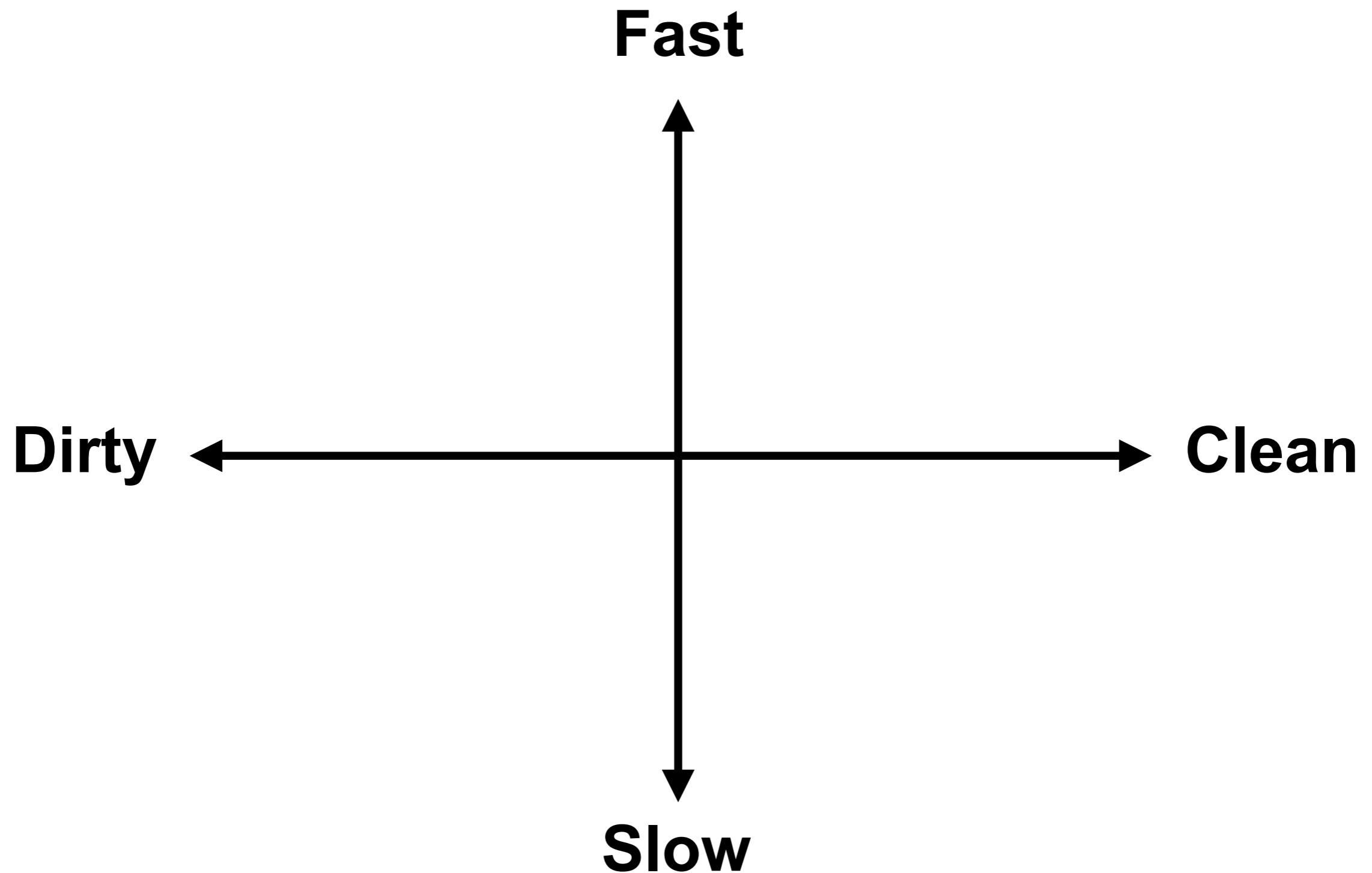
You're lazy

No time to be sloppy

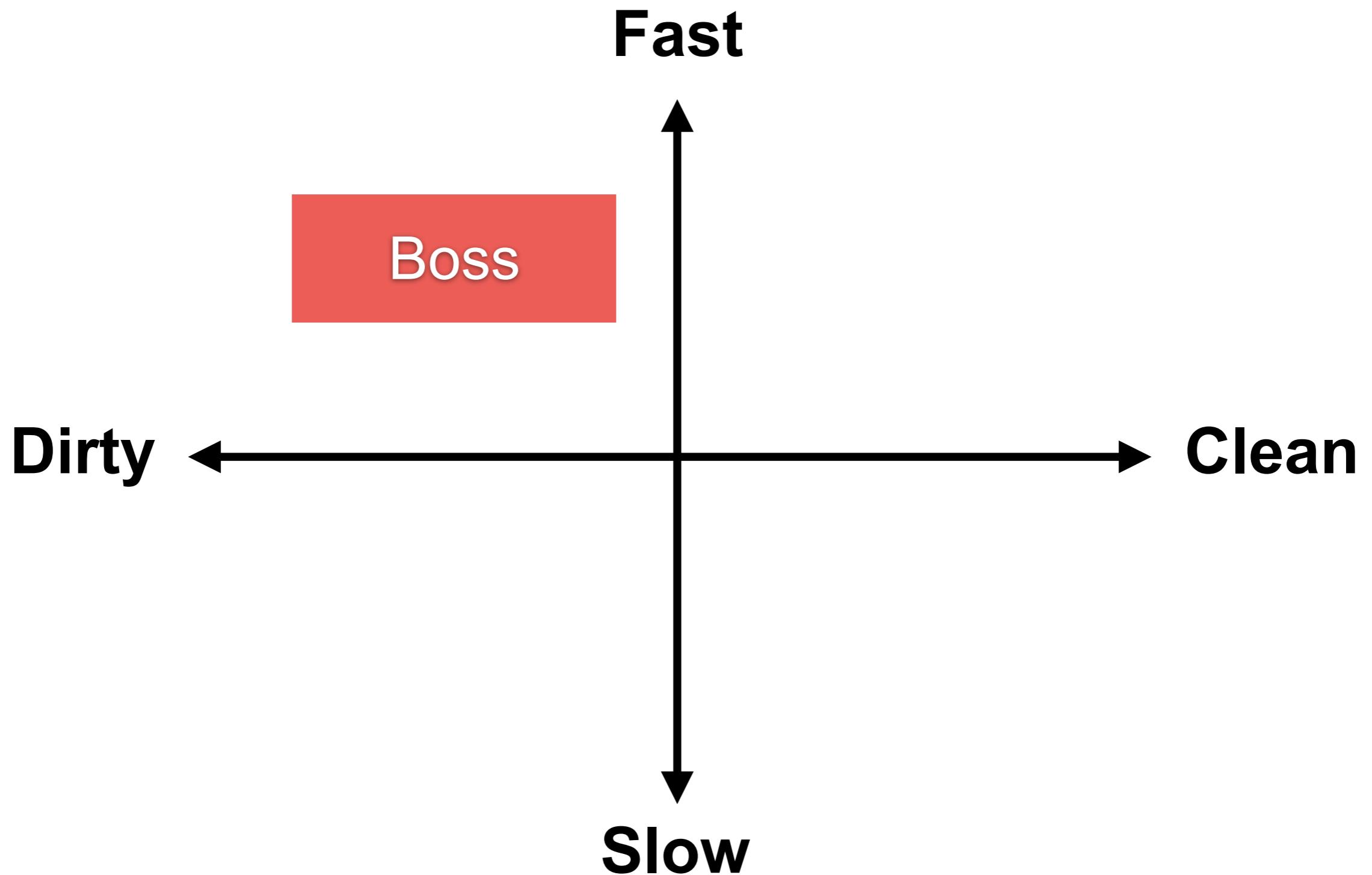
**It's foundation**



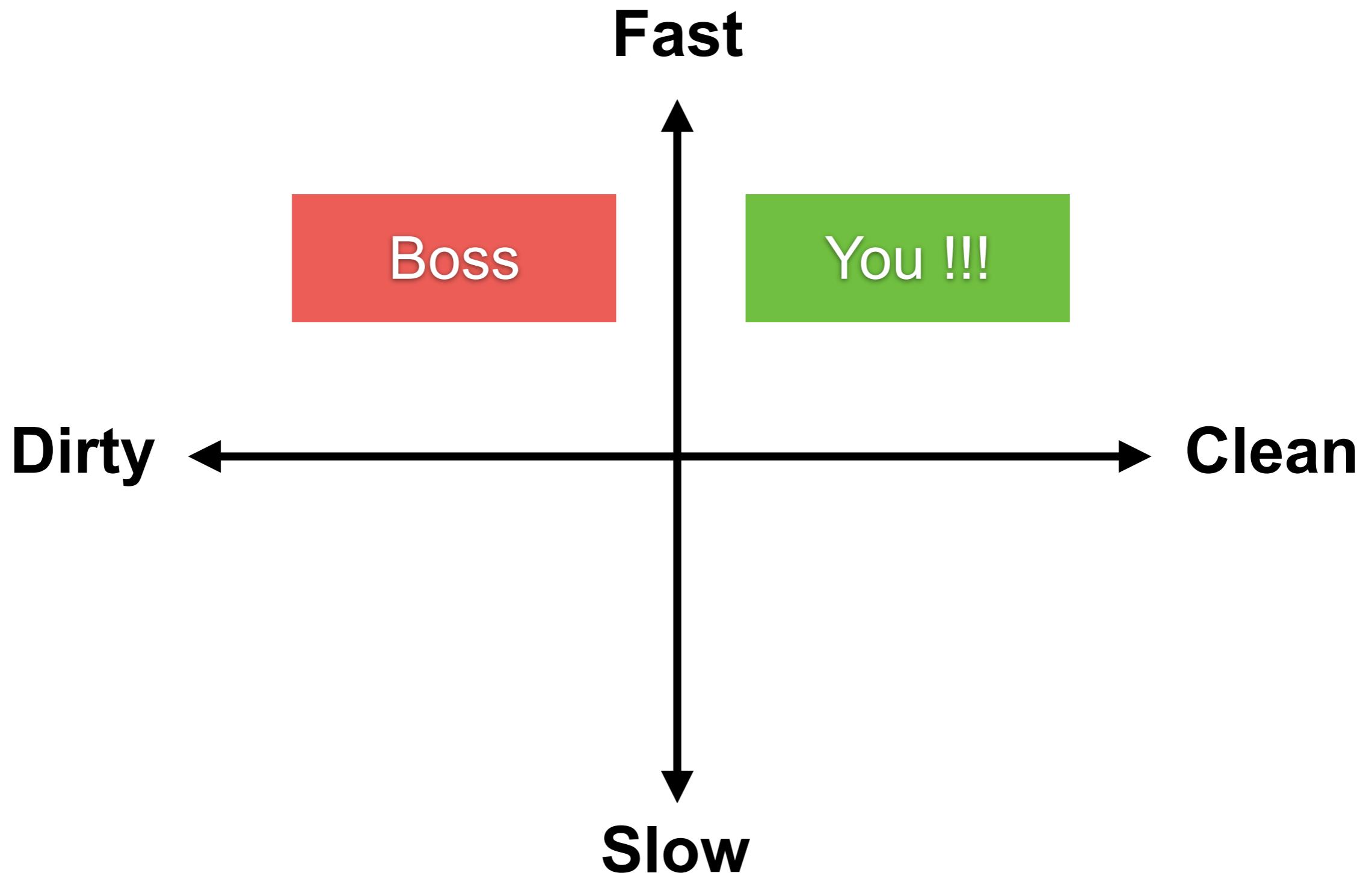
# Why should I care ?



# Why should I care ?



# Why should I care ?



# Foundation

Clean Code



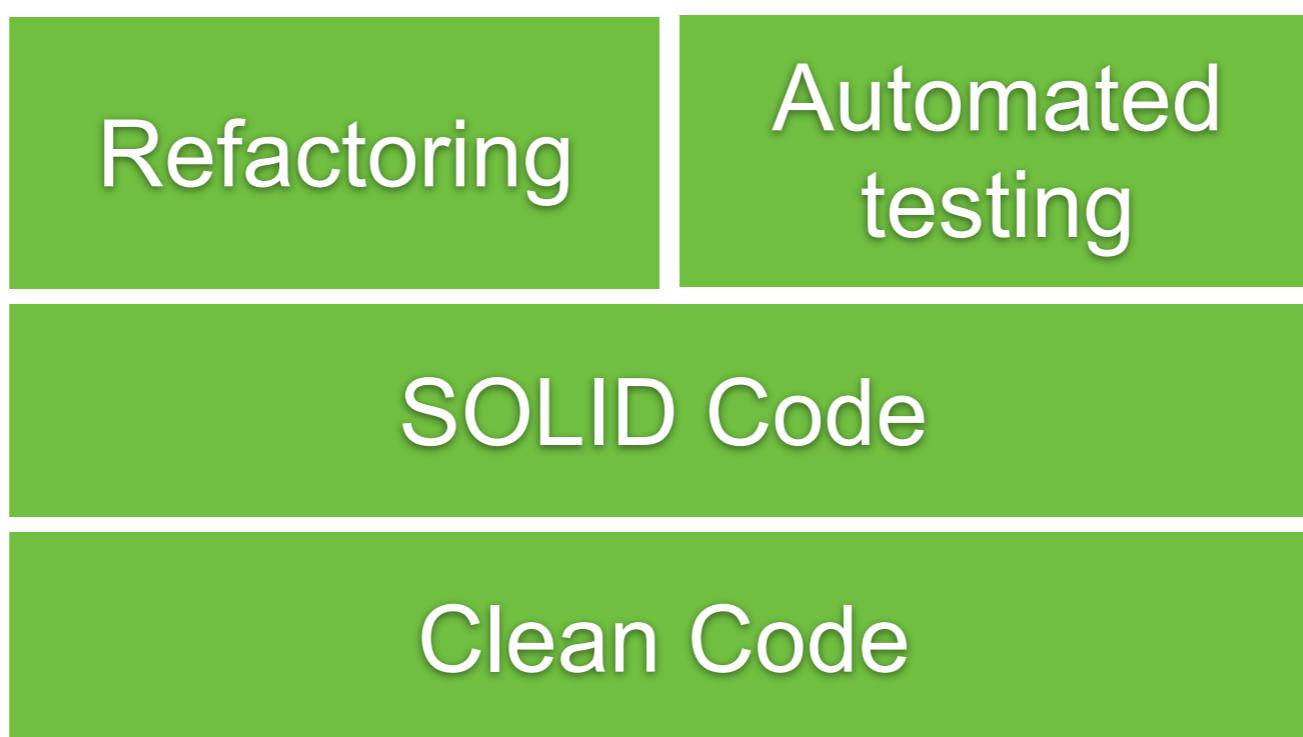
# Foundation

SOLID Code

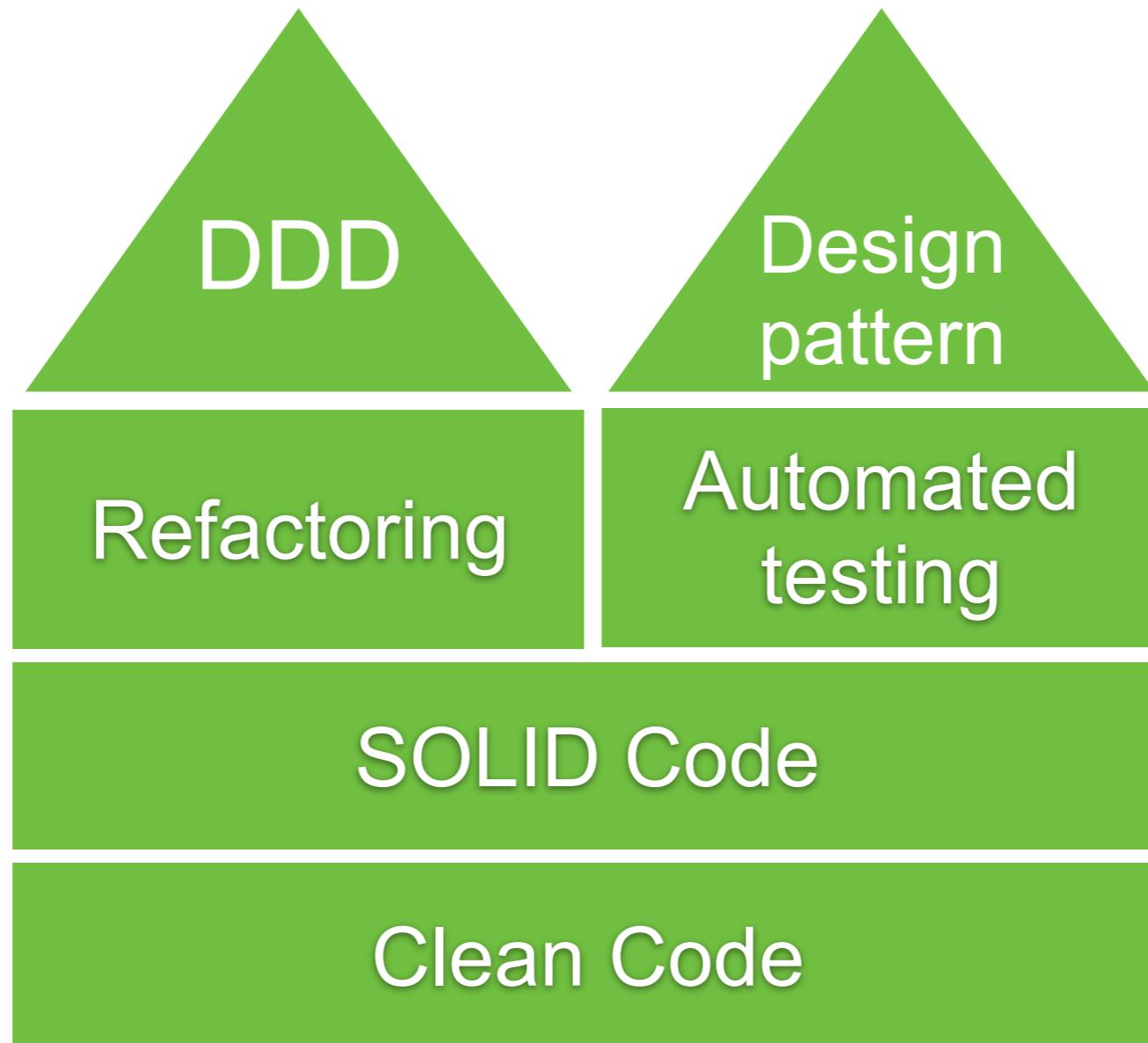
Clean Code



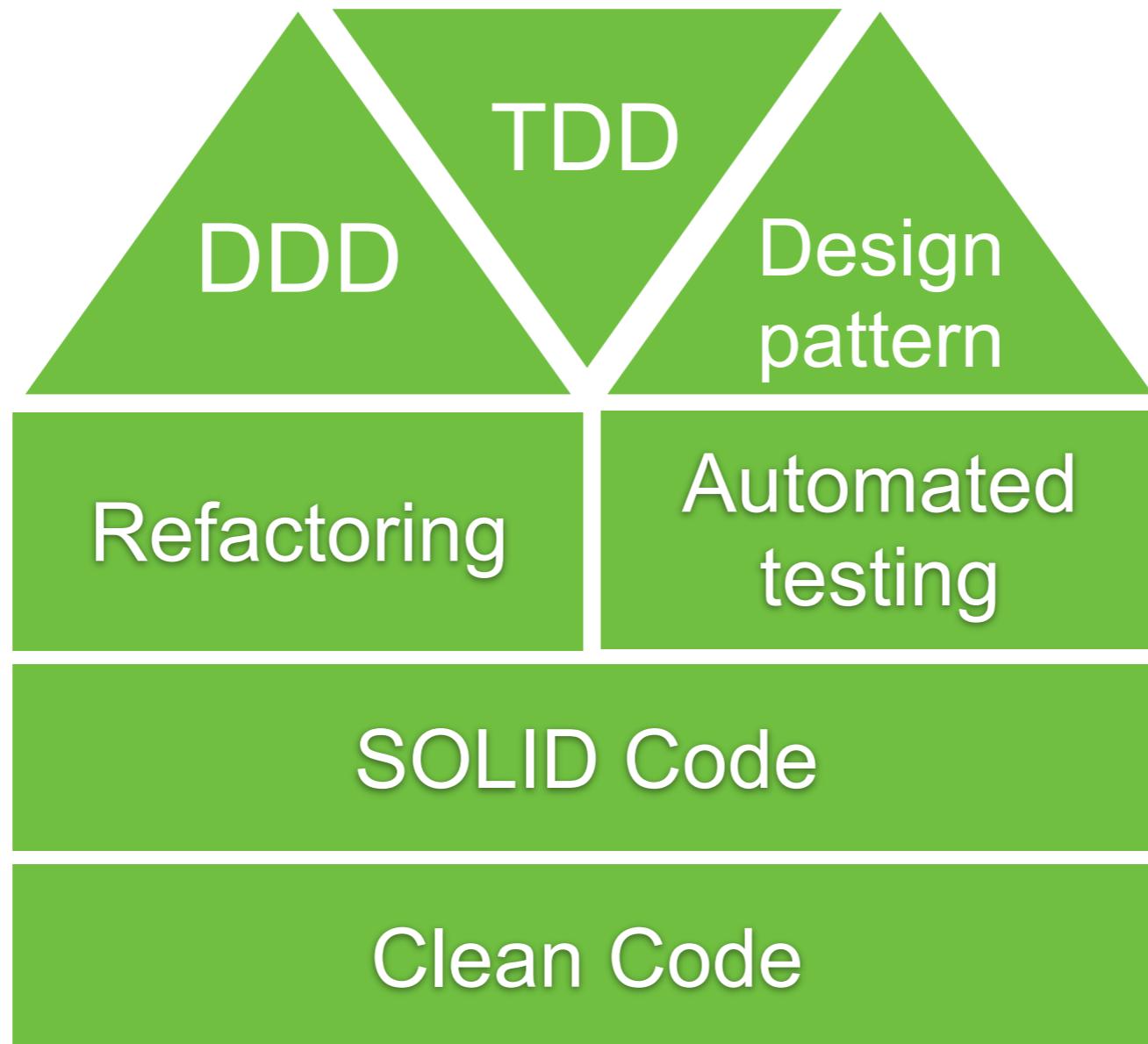
# Foundation



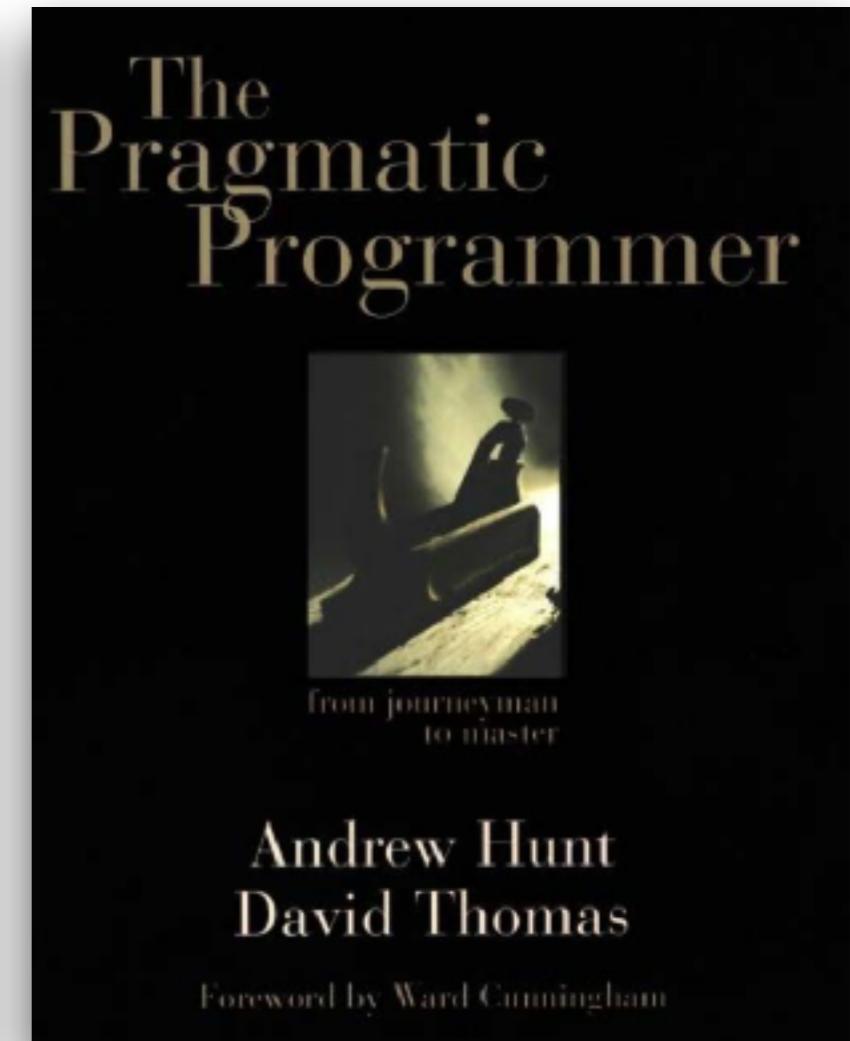
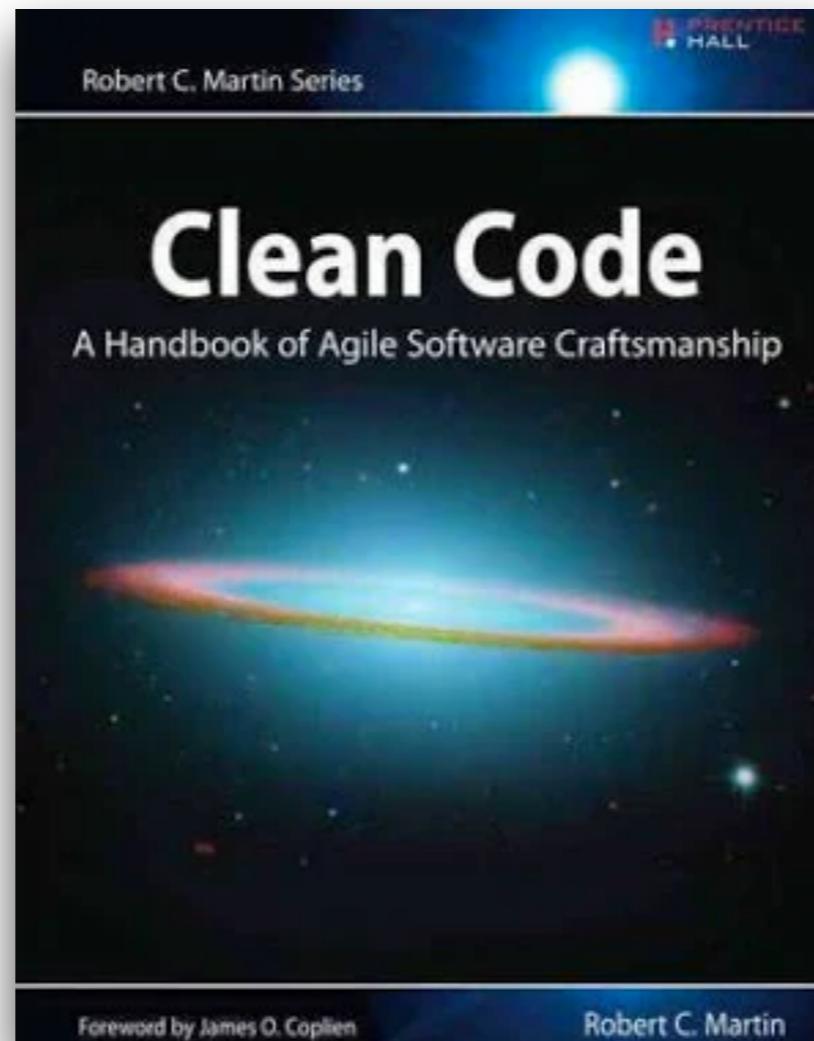
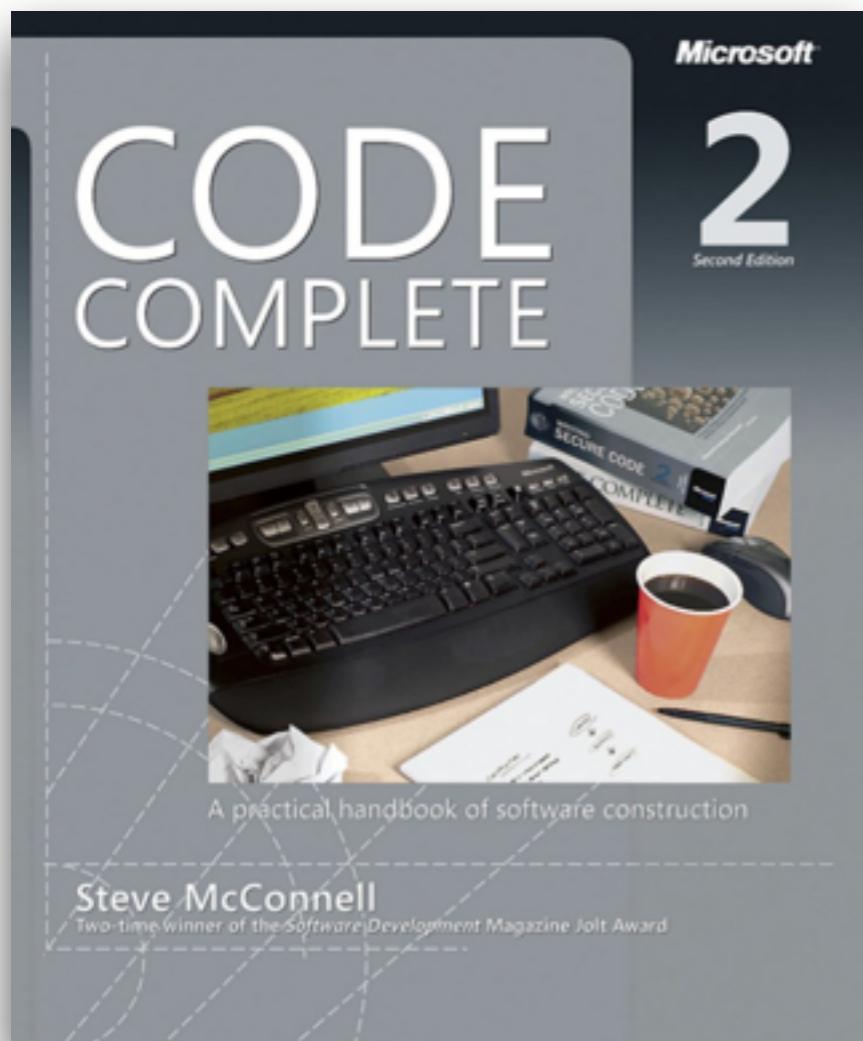
# Foundation



# Foundation



# Resources



# Principles for Clean Code

Right tool for the right job  
High signal to noise ratio  
Self-documenting



We should use

<^\_.^>

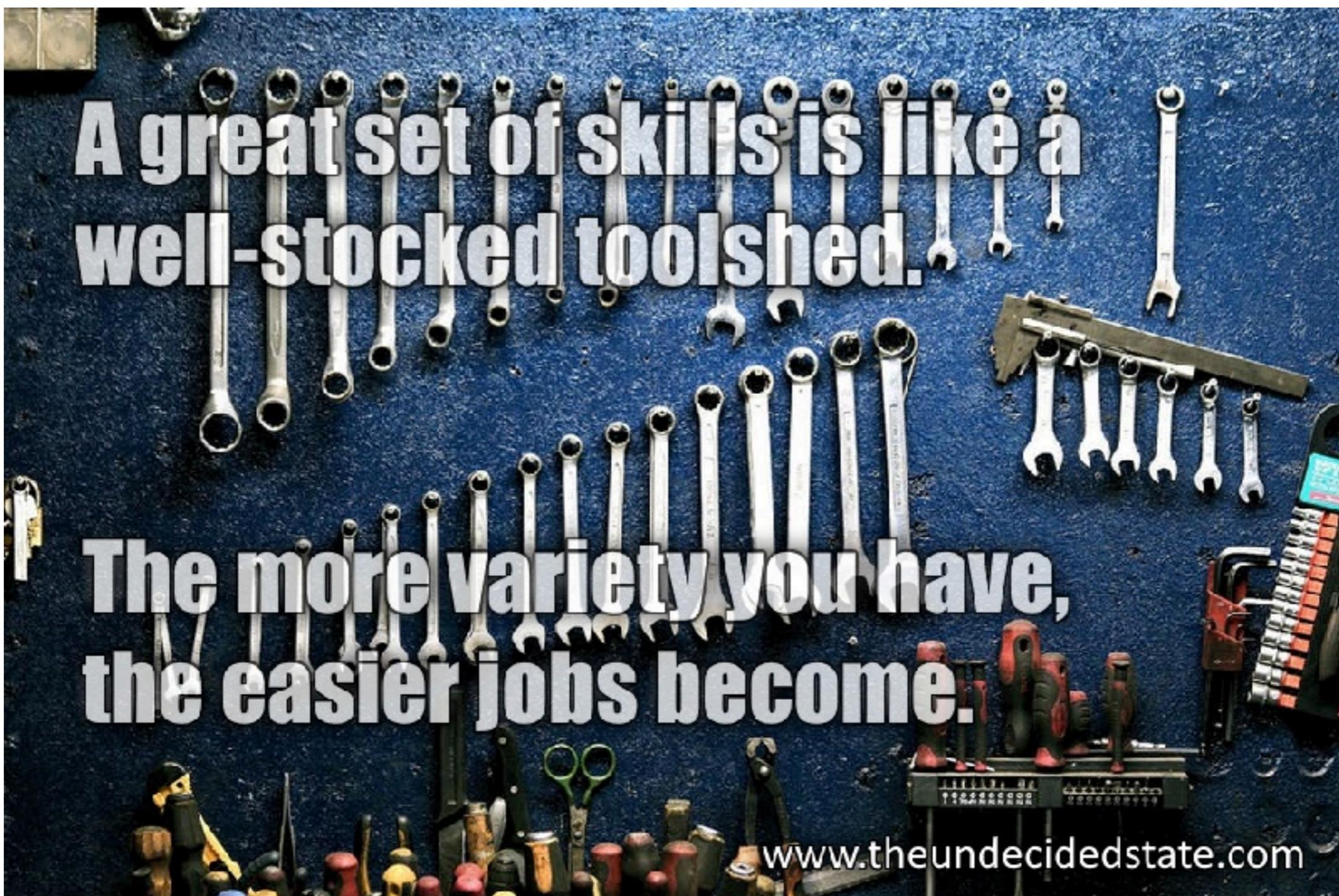
for everything !!



# Right tool for the right job



# Right tool for the right job



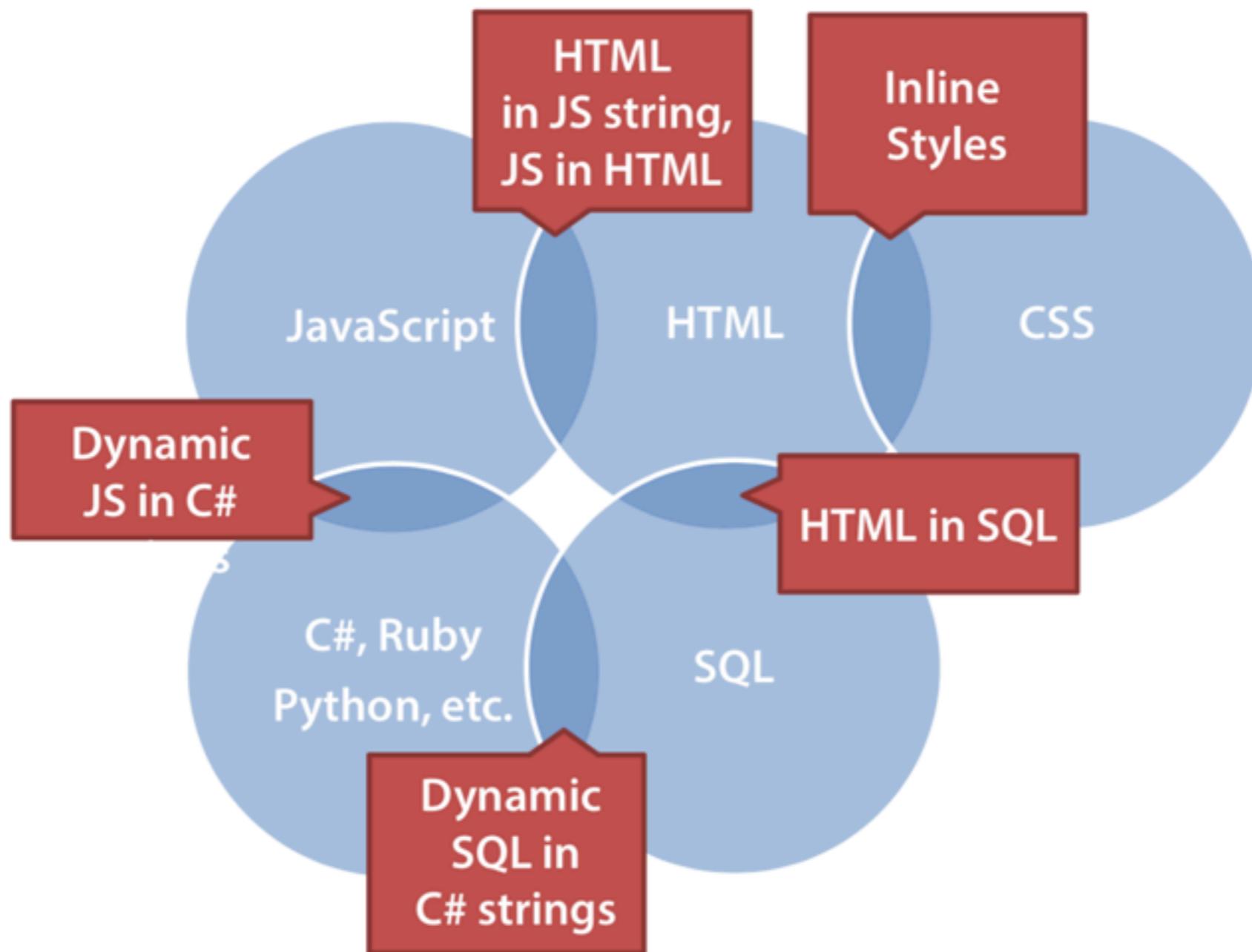
A great set of skills is like a  
well-stocked toolshed.

The more variety you have,  
the easier jobs become.

[www.theundecidedstate.com](http://www.theundecidedstate.com)



# Boundaries Matter



# Dirty

```
string script = @"<script type=""text/javascript"" defer=""defer"">
    //![CDATA[
        var _gaq = _gaq || [];
        _gaq.push(['_setAccount', '' + ws.GoogleAnalyticsID + ''']);
        _gaq.push(['_trackPageview']);

        (function() {
            var ga = document.createElement('script');
            ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
                '.google-analytics.com/ga.js';
            ga.setAttribute('async', 'true');
            document.documentElement.firstChild.appendChild(ga);
        })();
    //]]
</script>";
this.Header.Controls.Add(new LiteralControl("\r\n" + script));
```



# Clean

```
<!--In document head-->
<script type="text/javascript">
    var WebSiteSetup = { "GoogleAnalyticsKey": "JDSGI832JDUG9831" };
</script>

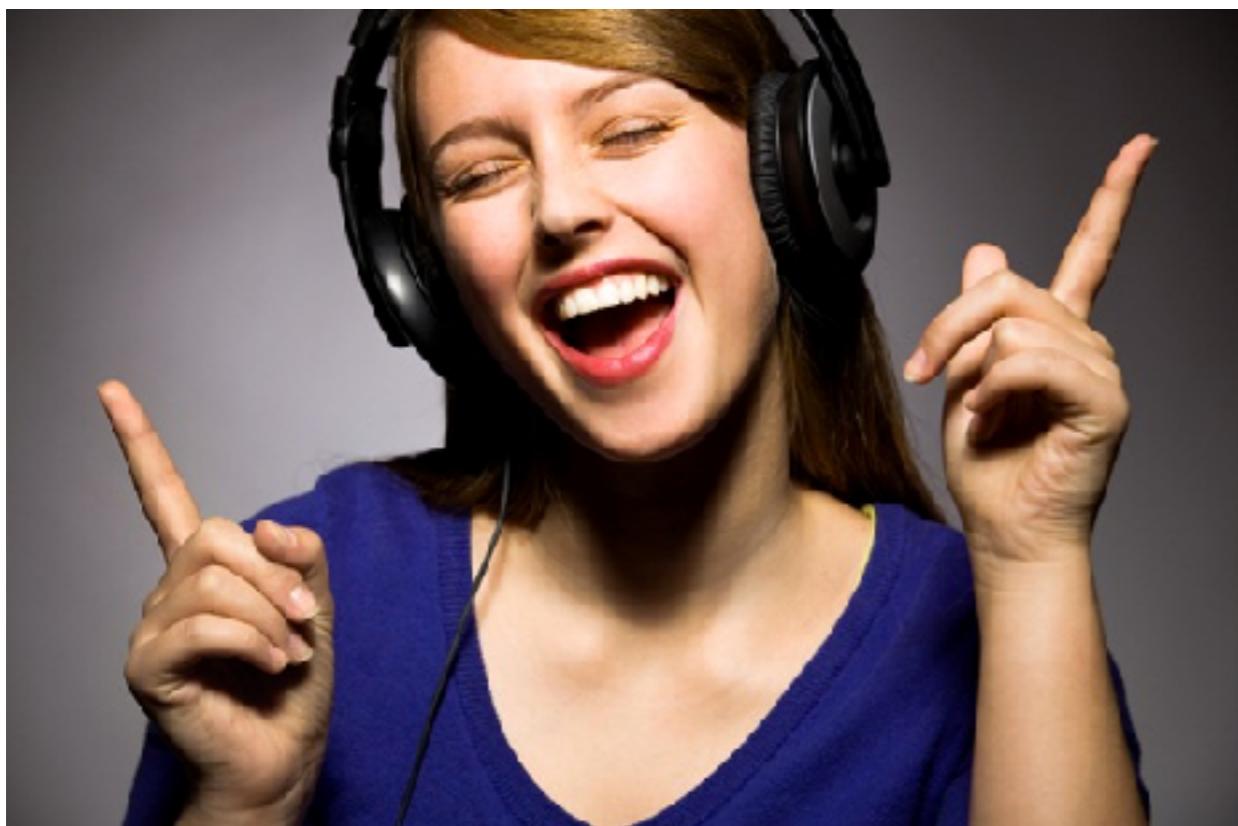
//In GoogleAnalytics.js
var _gaq = _gaq || [];
_gaq.push(['_setAccount', WebSiteSetup.GoogleAnalyticsKey]);
_gaq.push(['_trackPageview']);

(function () {
    var ga = document.createElement('script');
    ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') +
    '.google-analytics.com/ga.js';
    ga.setAttribute('async', 'true');
    document.documentElement.firstChild.appendChild(ga);
})();
```



# Maximize signal to noise ratio

**Signal**



**Noise**



# Signal => TED

Terse  
Expressive  
Do one thing



# Noise

High cyclomatic complexity

Huge classes

Long methods

Zombie code

Unnecessary comments

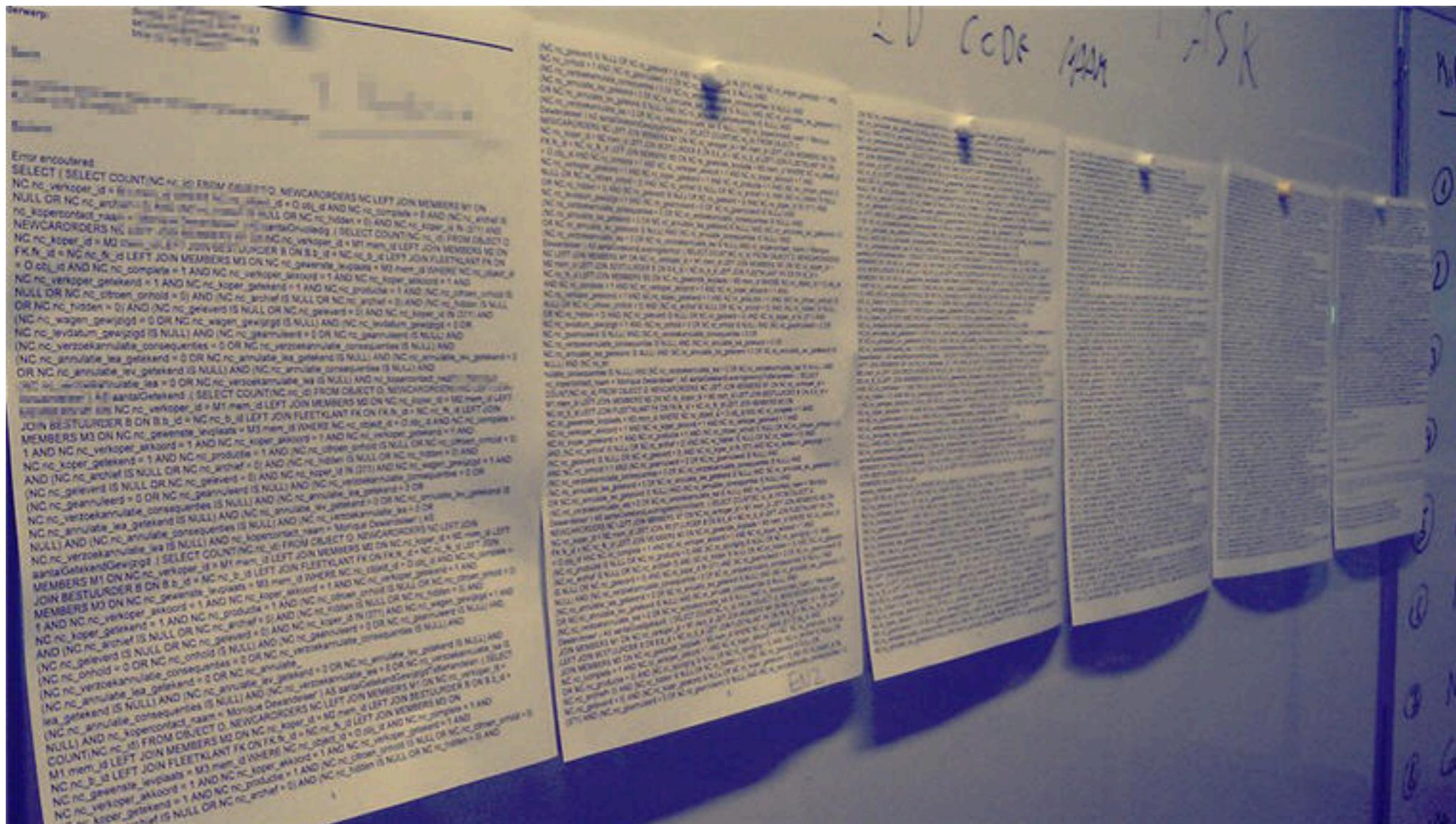
Poorly named structures

Duplication/Repetition

Excessive indentation



# SQL Query of the day !!



<https://thedailywtf.com/articles/The-Query-of-Despair>



# Concat string !!

```
return (((((((((((((((((((((((((((((((((((((  
((((((((((((((((((((((((((((((((((((((  
str + "" +  
this.Acct.Replace("", "") + "", "") + "" +  
this.AcctDist.ToString().Trim() + "", "") + "" +  
this.BatNbr.Replace("", "") + "", "") + "" +  
this.CmmnPct.ToString().Trim() + "", "") + "" +  
this.CnvFact.ToString().Trim() + "", "") + "" +  
this.ContractID.Replace("", "") + "", "") + "" +  
this.CostType.Replace("", "") + "", "") + "" +  
this.CpnyID.Replace("", "") + "", "") + "" +  
this.Crtd_DateTime.ToString().Trim().Replace("", "") + "", "") + "" +  
this.Crtd_Prog.Replace("", "") + "", "") + "" +  
this.Crtd_User.Replace("", "") + "", "") + "" +  
this.CuryExtCost.ToString().Trim() + "", "") + "" +  
this.CuryId.Replace("", "") + "", "") + "" +  
this.CuryMultDiv.Replace("", "") + "", "") + "" +  
this.CuryRate.ToString().Trim() + "", "") + "" +  
this.CuryTaxAmt00.ToString().Trim() + "", "") + "" +  
this.CuryTaxAmt01.ToString().Trim() + "", "") + "" +  
this.CuryTaxAmt02.ToString().Trim() + "", "") + "" +  
this.CuryTaxAmt03.ToString().Trim() + "", "") + "" +  
this.CuryTranAmt.ToString().Trim() + "", "") + "" +
```

<https://thedailywtf.com/articles/The-SQL-String>



# Code complexity

Cyclomatic Complexity	Risk Evaluation	Probability of Bad fix
1-10	Low risk, testable code	5%
11-20	Moderate Risk	10%
21-50	High Risk	30%
>50	Very High Risk, untestable code	40%



# Duplication is evil !!

DRY (Don't Repeat Yourself)

Copy and Paste if often a design problem



# Duplication issues

Decrease signal to noise ratio

Increases the number of LoC

Create a maintenance problem



# Look and find 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.Ma
            _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.Ma
            _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.Ma
            _DisclaimerUrls.Text += dl1_text + " ";
    }
}
```



# Self-documenting code

Well written code is self-documenting

Clear intent

Layer of abstraction

Format for readability

Code over Comments



# Naming Matters

```
List<decimal> p = new List<decimal>() { 5.50m, 10.48m, 12.69m };  
decimal t = 0;  
foreach (var i in p)  
{  
    t += i;  
}  
  
return t;
```



Could you read this book?

P was very angry with G for insulting her  
M. G kicked P in the A. He slept on the C.



# Better

```
List<decimal> prices = new List<decimal>() { 5.50m, 10.48m, 12.69m };
decimal total = 0;
foreach (var price in prices)
{
    total += price;
}

return total;
```



# Naming guidelines

Noun/Verb

Be specific

Single responsibility

Don't be cute

Frm Abbr !!



# Magic Number

## Dirty

```
if (age > 21)
{
    //body here
}
```

## Dirty

```
if (status == 2)
{
    //body here
}
```

## Clean

```
const int legalDrinkingAge = 21;
if (age > legalDrinkingAge)
{
    //body here
}
```

## Clean

```
if (status == Status.Active)
{
    //body here
}
```



# Complexity conditional

???

```
if (car.Year > 1980
    && (car.Make == "Ford" || car.Make == "Chevrolet")
    && car.Odometer < 100000
    && car.Vin.StartsWith("V2") || car.Vin.StartsWith("IA3"))
{
    //do lots of things here.
}
```



# Complexity conditional

Intermediate variables  
Encapsulate via function/method



# Intermediate variables

## Dirty

```
if (employee.Age > 55  
    && employee.YearsEmployed > 10 ← What question is this trying to answer?  
    && employee.IsRetired == true)  
{  
    //logic here  
}
```

## Clean

```
bool eligibleForPension = employee.Age > MinRetirementAge  
    && employee.YearsEmployed > MinPensionEmploymentYears  
    && employee.IsRetired;
```



# Encapsulate via function

## Dirty

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

Principle: Favor expressive code over comments



## Clean

```
if (ValidFileRequest(fileExtension, isActiveFile, isAdmin))

private bool ValidFileRequest(string fileExtension, bool isActiveFile, bool isAdmin)
{
    return (fileExtension == "mp4" ||
        fileExtension == "mpg" ||
        fileExtension == "avi")
        && (isAdmin || isActiveFile);
}
return validFileType && userIsAllowedToViewFile;
}
```



# Classic Problem ?

```
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;
    }
}
```



# Solution

Using polymorphism ?

```
public void LoginUser(User user)
{
    user.Login();
}
```

```
public abstract class User
{
    public string FirstName;
    public string LastName;
    public Status Status;
    public int AccountBalance;

    public abstract void Login();
}
```



# Solution

Using polymorphism ?

```
public class ActiveUser : User
{
    public override void Login()
    {
        //Active user logic here
    }
}
```

```
public class InactiveUser : User
{
    public override void Login()
    {
        //Inactive user logic here
    }
}
```

```
public class LockedUser : User
{
    public override void Login()
    {
        //Locked user logic here
    }
}
```



# Function/Method



# When to create a function ?



# When to create function ?

Duplication

Indentation

Unclear intent

>1 task



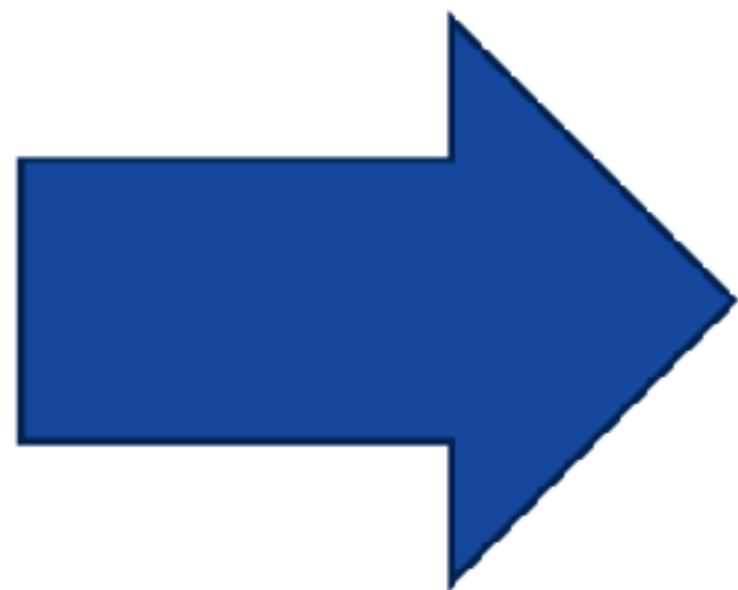
# Duplication

DRY (Don't Repeat Yourself)  
Code is a liability  
Less is more



# Excessive Indentation

Arrow code !!



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

*Comprehensive decreases beyond 3 levels of nested if blocks*



# 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
```

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

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



# 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;
}
```



**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” -**



# 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);
}
```



# Fail fast :: Avoid arrow code

Dirty !!

```
public void registerUser(String username, String password) {  
    if (username != null && username.trim().length() > 0) {  
        if (password != null && password.trim().length() > 0) {  
            // Register process  
  
        } else {  
            throw new RuntimeException("Password is required");  
        }  
    } else {  
        throw new RuntimeException("Username is required");  
    }  
}
```



# Fail fast :: Avoid arrow code

Clean

```
public void registerUser(String username, String password) {  
    if (username != null && username.trim().length() > 0){  
        throw new RuntimeException("Username is required");  
    }  
    if (password != null && password.trim().length() > 0){  
        throw new RuntimeException("Password is required");  
    }  
    // Register process  
}
```



# Do one thing

Aids the reader

Promotes reuse

Ease naming and testing

Avoid side-effect



# How many parameters ?

**0 - 2 parameters**

Easier to understand

Easier to test

Help assure function/method does one thing



# How many parameters ?

**Dirty**

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

**Clean**

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



# Flag arguments

A sign the function/method is doing 2 things

**Dirty**

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

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



# Flag arguments

A sign the function/method is doing 2 things

## Clean

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

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



# Signs it's too long ?

Whitespace  
and comment

Scrolling  
required

Naming issues

Multiple  
conditional

Hard to  
understand



Rarely be over 20 lines  
Hardly ever over 100 lines  
No more than 3 parameters

*- Robert C. Martin , “Clean Code” -*



**Simple function can be longer.  
Complex function should be short.**

**- *Linux style guide* -**



# Exception handling



# 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
}
```



# Stay clean





# When to refactor to clean code

You need to work with the code

Difficult to change

Test coverage to protect from regression



# Accept no broken windows



# Code review

Promote proactive cleanliness

Set guidelines

Assure readability





© J.T.PRESTA  
1/30/2013



# Working in pairs

Real-time code review

Increase quality

Naming and refactoring is easier



# Lunch and Learn

Showcase what you have learned

Be a thought leader

Spark interest in others

Get free food !!



# The boy scout rule

Always leave the code you're editing a little better than you found it.

*- Robert C. Martin -*



Clean code always look like it was written  
by someone who **cares**

*- Michael Feathers -*



# Workshop

## Refactoring legacy code by simple design



# Clone code in IDE

<https://github.com/up1/workshop-refactoring-legacy>

The screenshot shows a GitHub repository page. At the top, the repository name 'up1 / workshop-refactoring-legacy' is displayed, along with statistics: 1 commit, 1 branch, 0 releases, and 1 contributor. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a prominent green 'Clone or download' button. The main area shows a list of files: 'up1 Starter code of workshop' (latest commit 55cbe4b 6 minutes ago), 'src/main/java/demo/legacy' (Starter code of workshop, 6 minutes ago), and 'pom.xml' (Starter code of workshop, 6 minutes ago).



# Simple Design

1. Pass all the tests
2. Reveal intent
3. No duplication/ DRY
4. Small/fewer parts



# Rule of legacy code refactoring

Don't break it



# Code overview



# **Rule 4 :: Small/fewer parts**

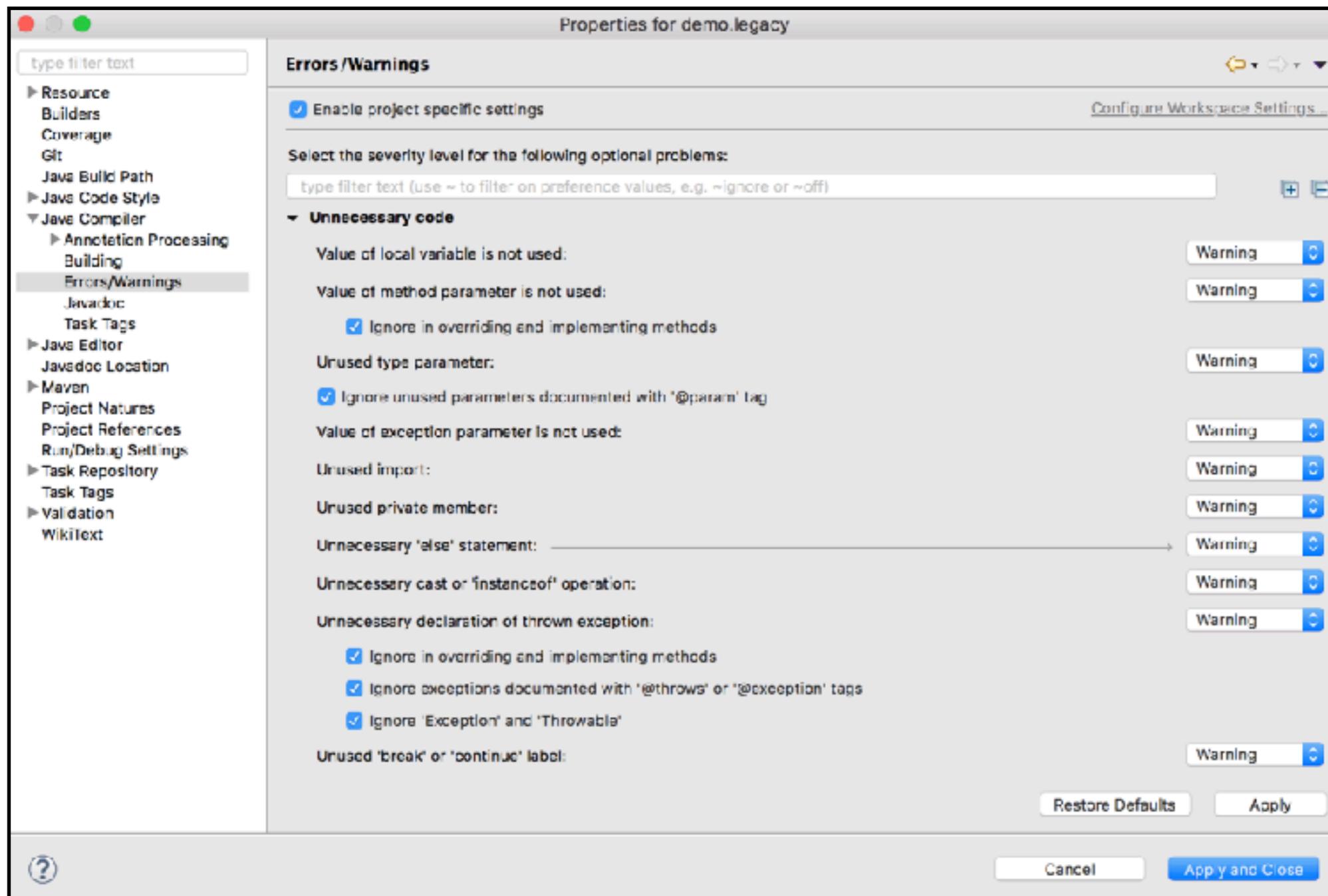


# Find and remove dead code



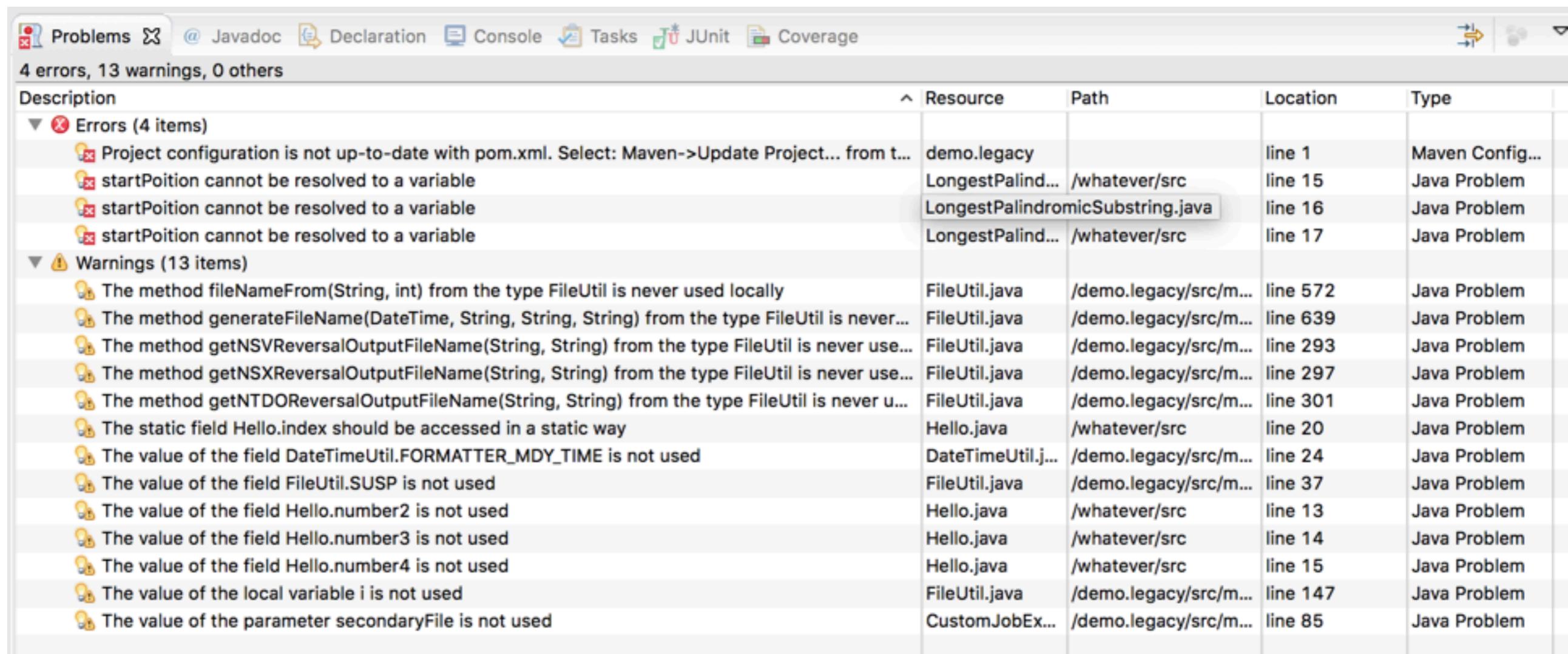
# Eclipse IDE

## Java compiler -> Errors and Warnings



# Eclipse IDE

## See all warnings and fix



The screenshot shows the Eclipse IDE's Problems view. The title bar includes tabs for Problems, Javadoc, Declaration, Console, Tasks, JUnit, and Coverage. Below the tabs, it displays "4 errors, 13 warnings, 0 others". The main area is a table with columns: Description, Resource, Path, Location, and Type. The Description column is expanded to show two sections: Errors (4 items) and Warnings (13 items). The errors section contains four entries, each with a red error icon and a message about Maven configuration or variable resolution issues. The warnings section contains 13 entries, each with a yellow warning icon and a message about unused code or static access issues.

Description	Resource	Path	Location	Type
Errors (4 items)				
Project configuration is not up-to-date with pom.xml. Select: Maven->Update Project... from t...	demo.legacy		line 1	Maven Config...
startPoition cannot be resolved to a variable	LongestPalind...	/whatever/src	line 15	Java Problem
startPoition cannot be resolved to a variable	LongestPalindromicSubstring.java		line 16	Java Problem
startPoition cannot be resolved to a variable	LongestPalind...	/whatever/src	line 17	Java Problem
Warnings (13 items)				
The method fileNameFrom(String, int) from the type FileUtils is never used locally	FileUtil.java	/demo.legacy/src/m...	line 572	Java Problem
The method generateFileName(DateTime, String, String, String) from the type FileUtils is never...	FileUtil.java	/demo.legacy/src/m...	line 639	Java Problem
The method getNSVReversalOutputFileName(String, String) from the type FileUtils is never use...	FileUtil.java	/demo.legacy/src/m...	line 293	Java Problem
The method getNSXReversalOutputFileName(String, String) from the type FileUtils is never use...	FileUtil.java	/demo.legacy/src/m...	line 297	Java Problem
The method getNTDOReversalOutputFileName(String, String) from the type FileUtils is never u...	FileUtil.java	/demo.legacy/src/m...	line 301	Java Problem
The static field Hello.index should be accessed in a static way	Hello.java	/whatever/src	line 20	Java Problem
The value of the field DateTimeUtil.FORMATTER_MDY_TIME is not used	DateTimeUtil.j...	/demo.legacy/src/m...	line 24	Java Problem
The value of the field FileUtils.SUSP is not used	FileUtil.java	/demo.legacy/src/m...	line 37	Java Problem
The value of the field Hello.number2 is not used	Hello.java	/whatever/src	line 13	Java Problem
The value of the field Hello.number3 is not used	Hello.java	/whatever/src	line 14	Java Problem
The value of the field Hello.number4 is not used	Hello.java	/whatever/src	line 15	Java Problem
The value of the local variable i is not used	FileUtil.java	/demo.legacy/src/m...	line 147	Java Problem
The value of the parameter secondaryFile is not used	CustomJobEx...	/demo.legacy/src/m...	line 85	Java Problem



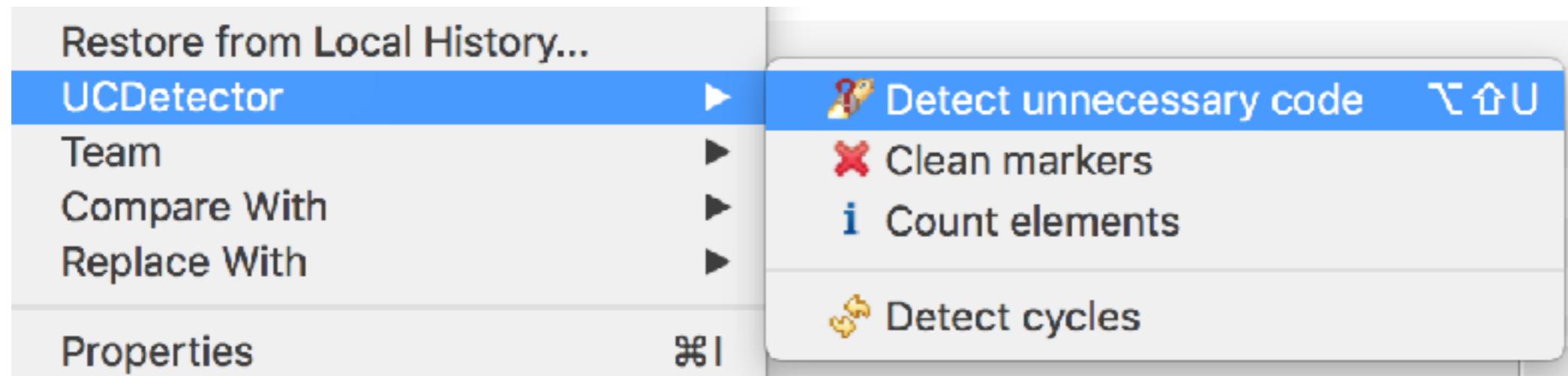
# **Find and remove more dead code with ucdetector**

<http://www.ucdetector.org/>



# Eclipse IDE

Run UCDetector => Detect unnecessary code



# Eclipse IDE

# More warning and fix !!!

Description	Resource	Path
▶ <span style="color: red;">X</span> Errors (4 items)		
▼ <span style="color: orange;">!</span> Warnings (100 of 149 items)		
<span style="color: orange;">!</span> Change visibility of Class "DateTimeUtil" to default - May cause compile errors!	DateTimeUtil.j...	/demo.le...
<span style="color: orange;">!</span> Change visibility of Class "FileUtil" to default - May cause compile errors!	FileUtil.java	/demo.le...
<span style="color: orange;">!</span> Change visibility of Constant "FileUtil.DATE_PATTERN" to private	FileUtil.java	/demo.le...
<span style="color: orange;">!</span> Change visibility of Constant "FileUtil.OPM_PROCESS" to private	FileUtil.java	/demo.le...
<span style="color: orange;">!</span> Change visibility of Constant "FileUtil.PAM_PROCESS" to private	FileUtil.java	/demo.le...
<span style="color: orange;">!</span> Change visibility of Constant "FileUtil.RRB_PROCESS" to private	FileUtil.java	/demo.le...
<span style="color: orange;">!</span> Change visibility of Interface "Constants" to default - May cause compile errors!	Constants.java	/demo.le...
<span style="color: orange;">!</span> Change visibility of Method "DateTimeUtil.convertToLocalDate(java.util.Date)" to default	DateTimeUtil.j...	/demo.le...
<span style="color: orange;">!</span> Change visibility of Method "DateTimeUtil.formatDateMdyNoSlashes(java.util.Date)" to default	DateTimeUtil.j...	/demo.le...
<span style="color: orange;">!</span> Change visibility of Method "DateTimeUtil.formatDateWithPattern(LocalDate, String)" to default	DateTimeUtil.j...	/demo.le...
<span style="color: orange;">!</span> Change visibility of Method "DateTimeUtil.formatDateYmdNoDashes(LocalDate)" to default	DateTimeUtil.j...	/demo.le...
<span style="color: orange;">!</span> Change visibility of Method "DateTimeUtil.isBusinessDate(LocalDate, List<LocalDate>)" to private	DateTimeUtil.j...	/demo.le...
<span style="color: orange;">!</span> Change visibility of Method "DateTimeUtil.isWeekend(LocalDate)" to private	DateTimeUtil.j...	/demo.le...
<span style="color: orange;">!</span> Change visibility of Method "DateTimeUtil.parseLocalDateFrom(String)" to default	DateTimeUtil.j...	/demo.le...
<span style="color: orange;">!</span> Change visibility of Method "DateTimeUtil.parseLocalDateFrom(String, Locale)" to default	DateTimeUtil.j...	/demo.le...



# 1. Pass all the test !!!



# Start to write tests

