Der Sprecher

Hendrik Lösch

Senior Consultant

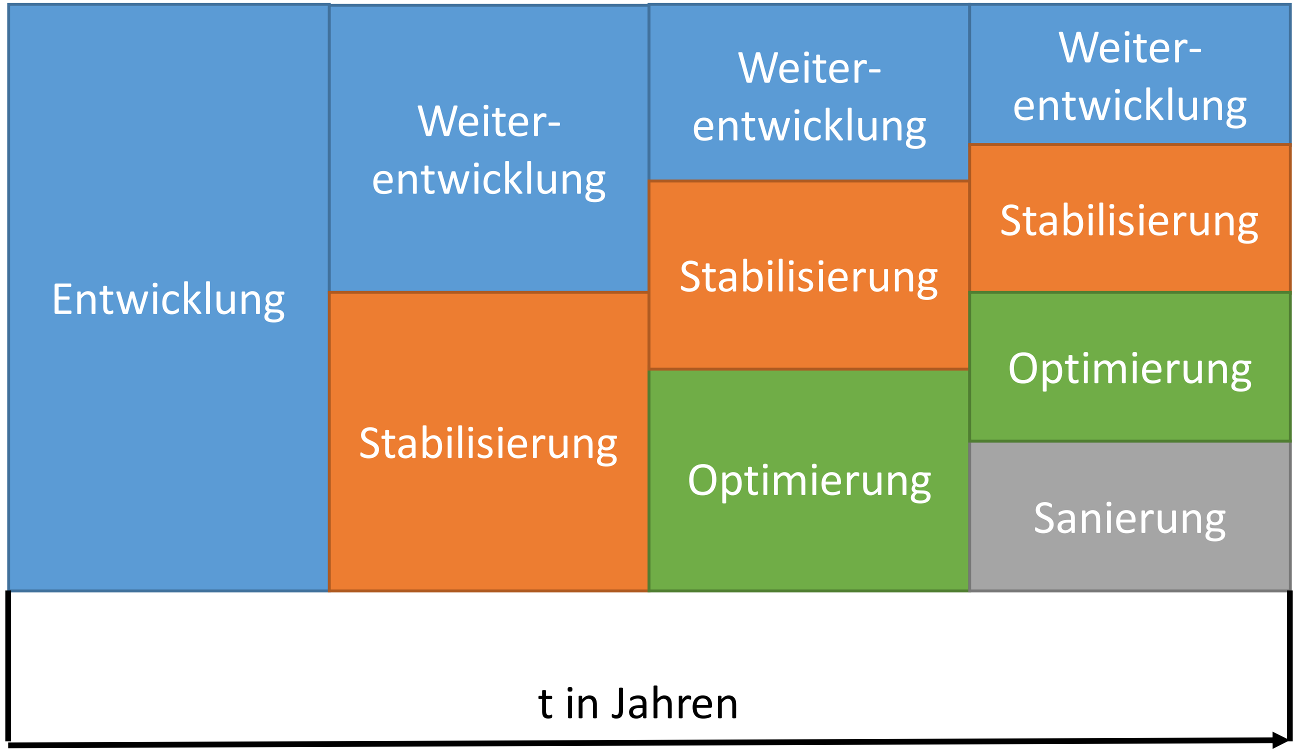
Coach

@HerrLoesch

Hendrik.Loesch@saxsys.de

Just-About.Net

Evolutionäre Softwareentwicklung



Entwicklung

Weiterentwicklung

Stabilisierung

Weiterentwicklung

Stabilisierung

Optimierung

Weiterentwicklung

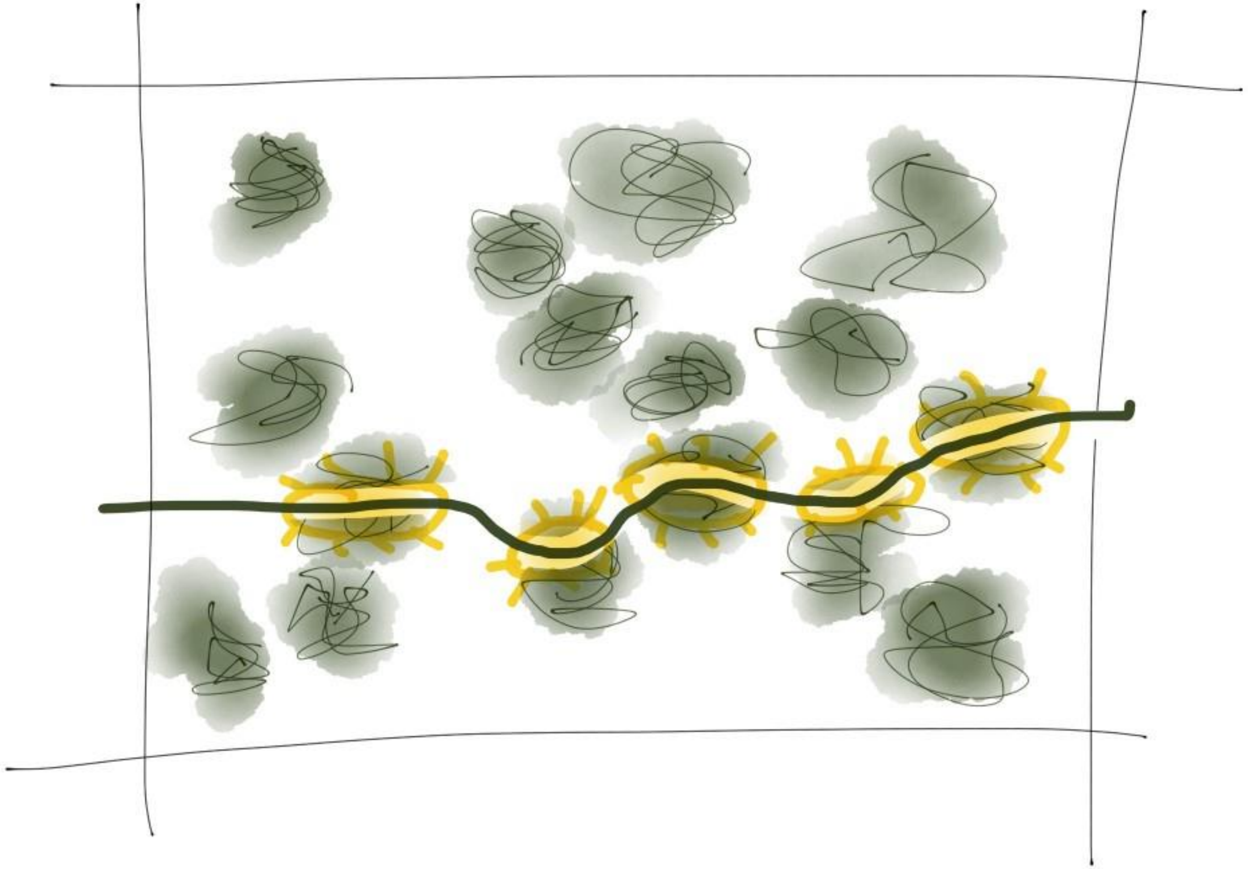
Stabilisierung

Optimierung

Sanierung

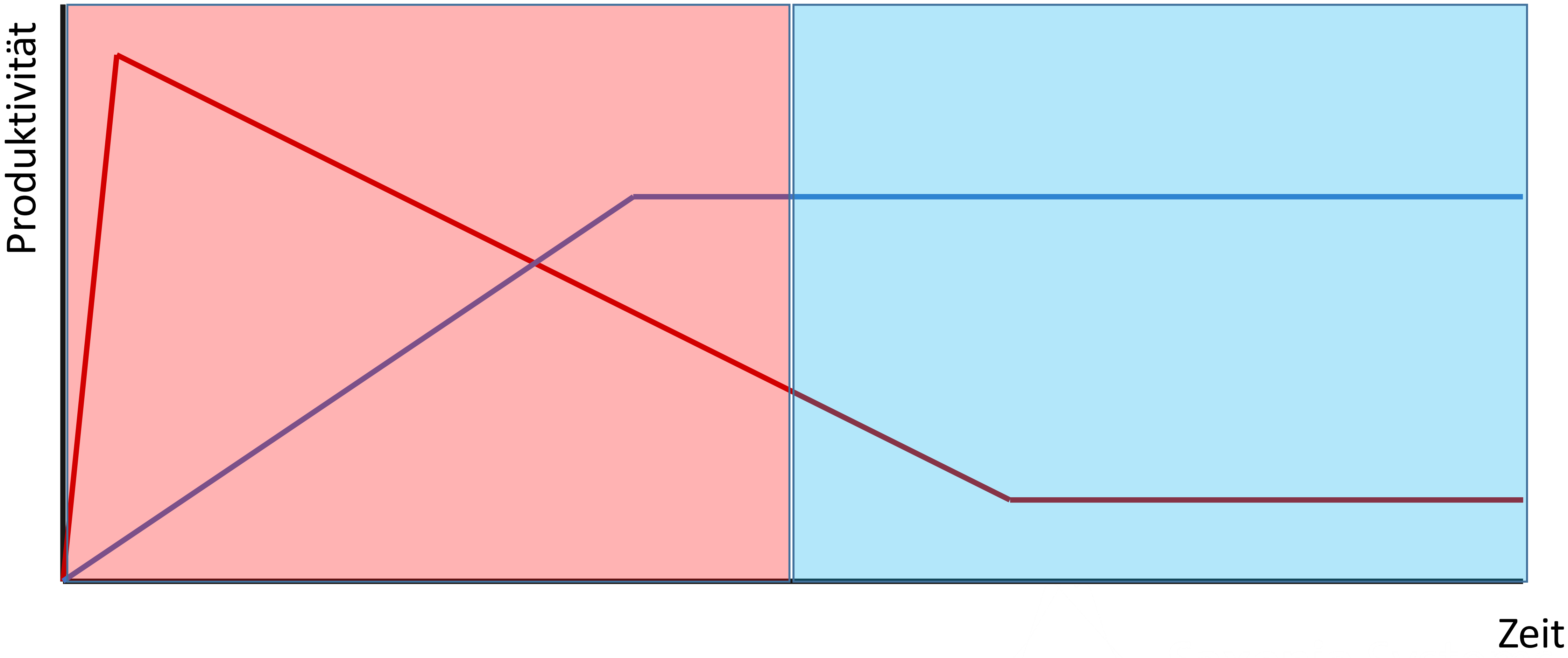
t in Jahren

Technische Schuld

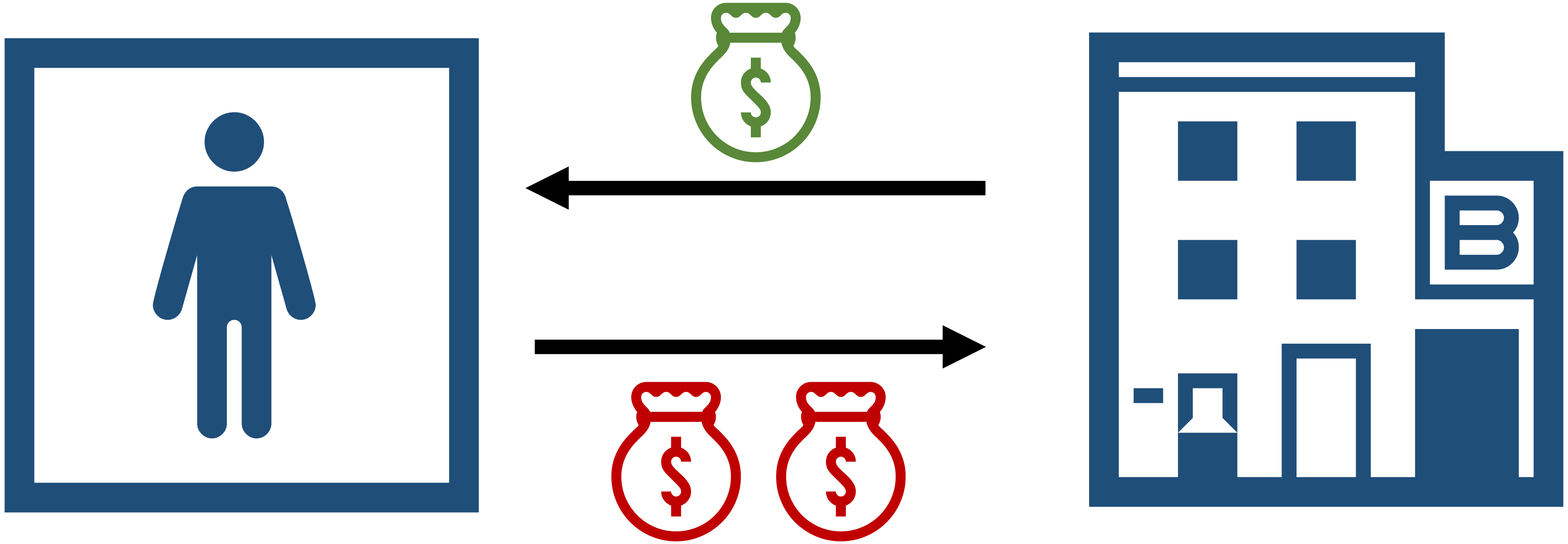


Quelle: Ron Jeffries http://xprogramming.com/articles/refactoring-not-on-the-backlog

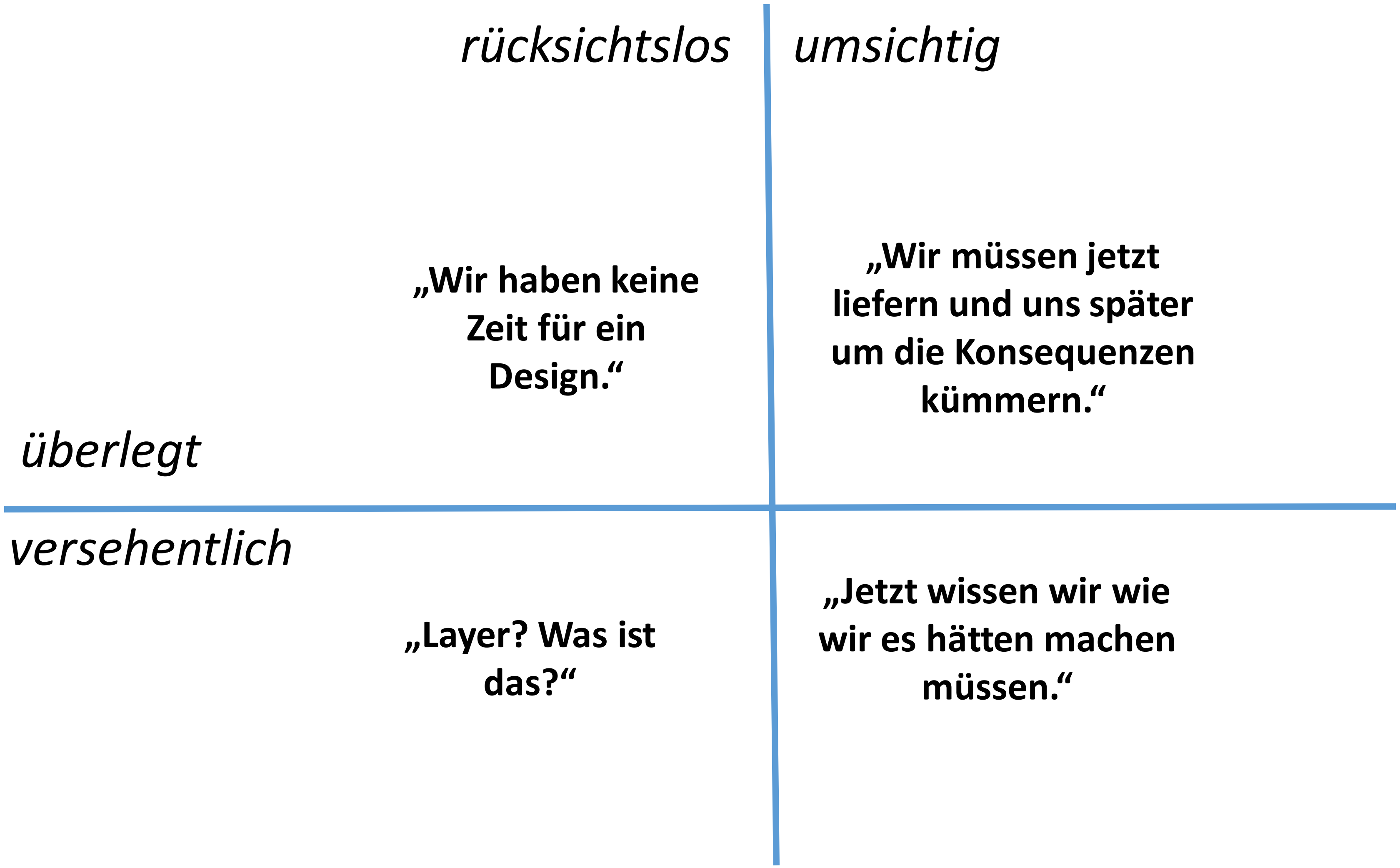
Technische Schuld



Technische Schuld



Technische Schuld



rücksichtslos

umsichtig

überlegt

versehentlich

„Wir haben keine Zeit für ein Design.“

„Wir müssen jetzt liefern und uns später um die Konsequenzen kümmern.“

„Layer? Was ist das?“

„Jetzt wissen wir wie wir es hätten machen müssen.“

Quelle: Martin Fowler http://martinfowler.com/bliki/TechnicalDebtQuadrant.html

Coding Rules

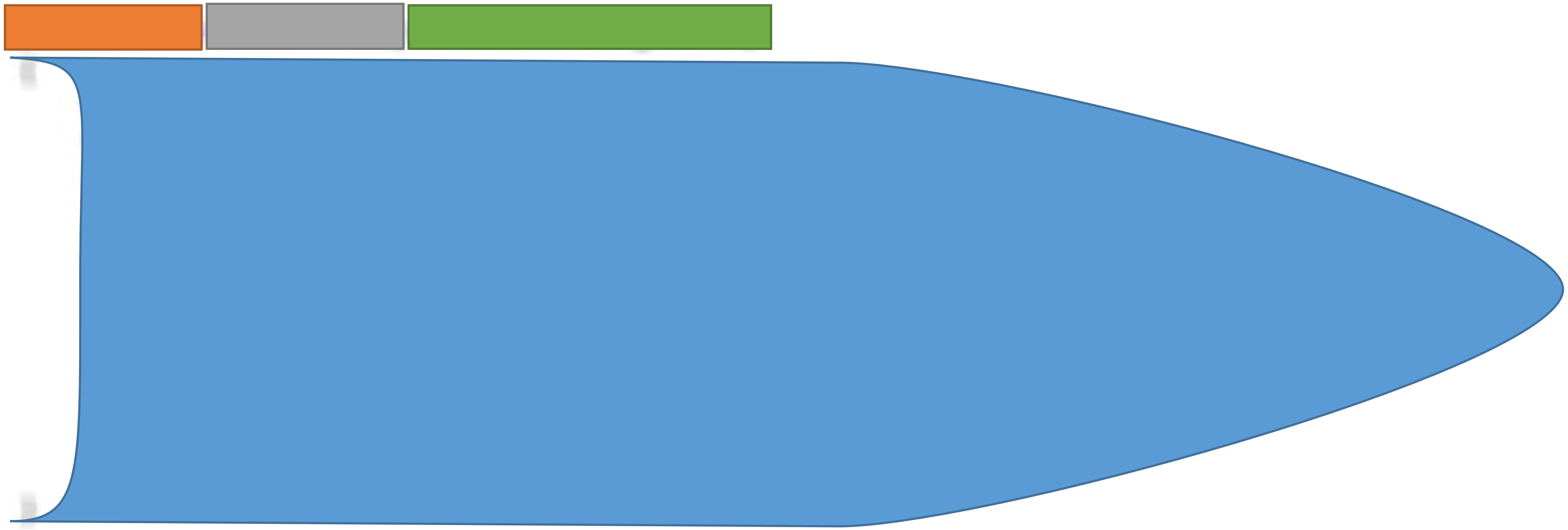
Lesbarkeit & Code Stil

Gmäes enia Shutidce ebneir älgnihcesn Uveinihnstert ißt es nchit whcitich, (...)

Gmäeß eneir Sutide eneir elgnihcesn Uvinisterät ist es nchit witihcg, (…)

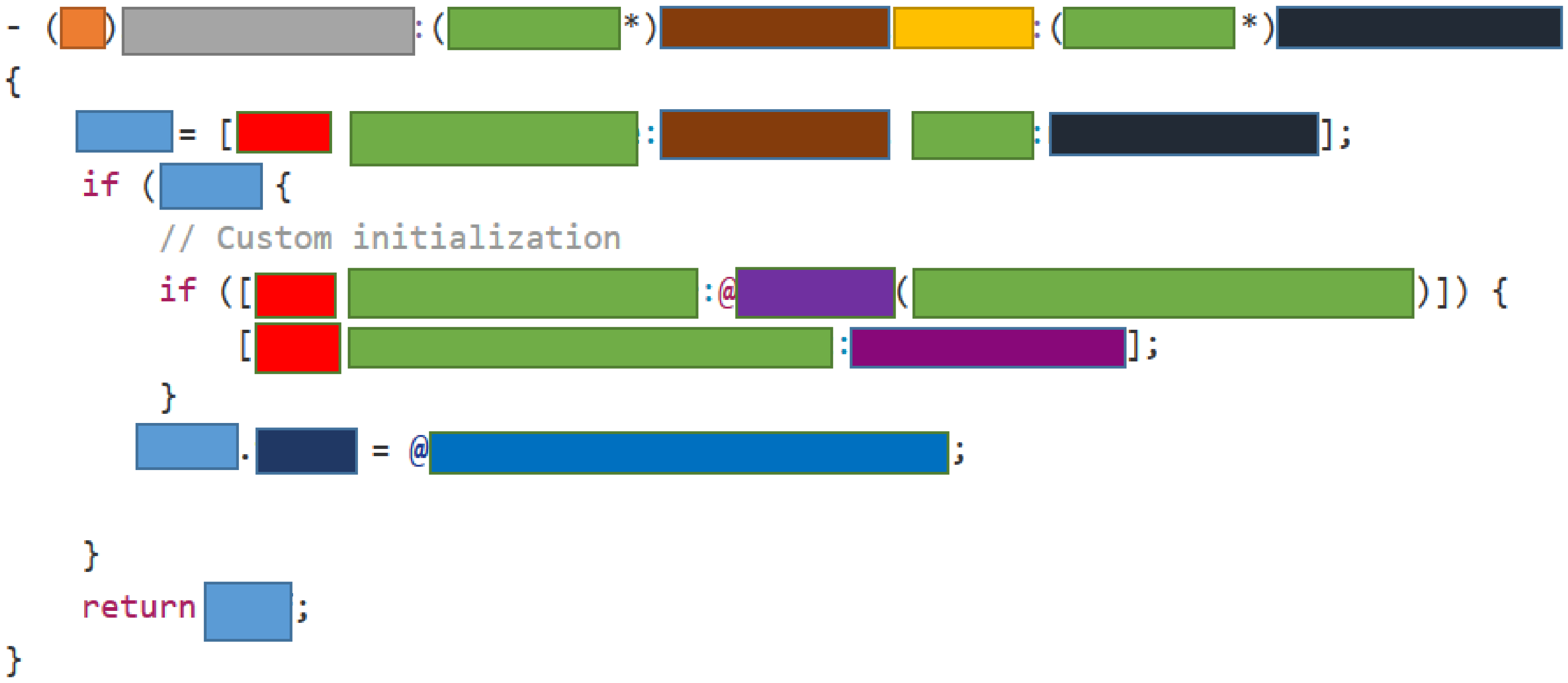
Quelle: http://www.fehler-haft.de/wissen/buchstabensalat.html

Lesbarkeit & Code Stil



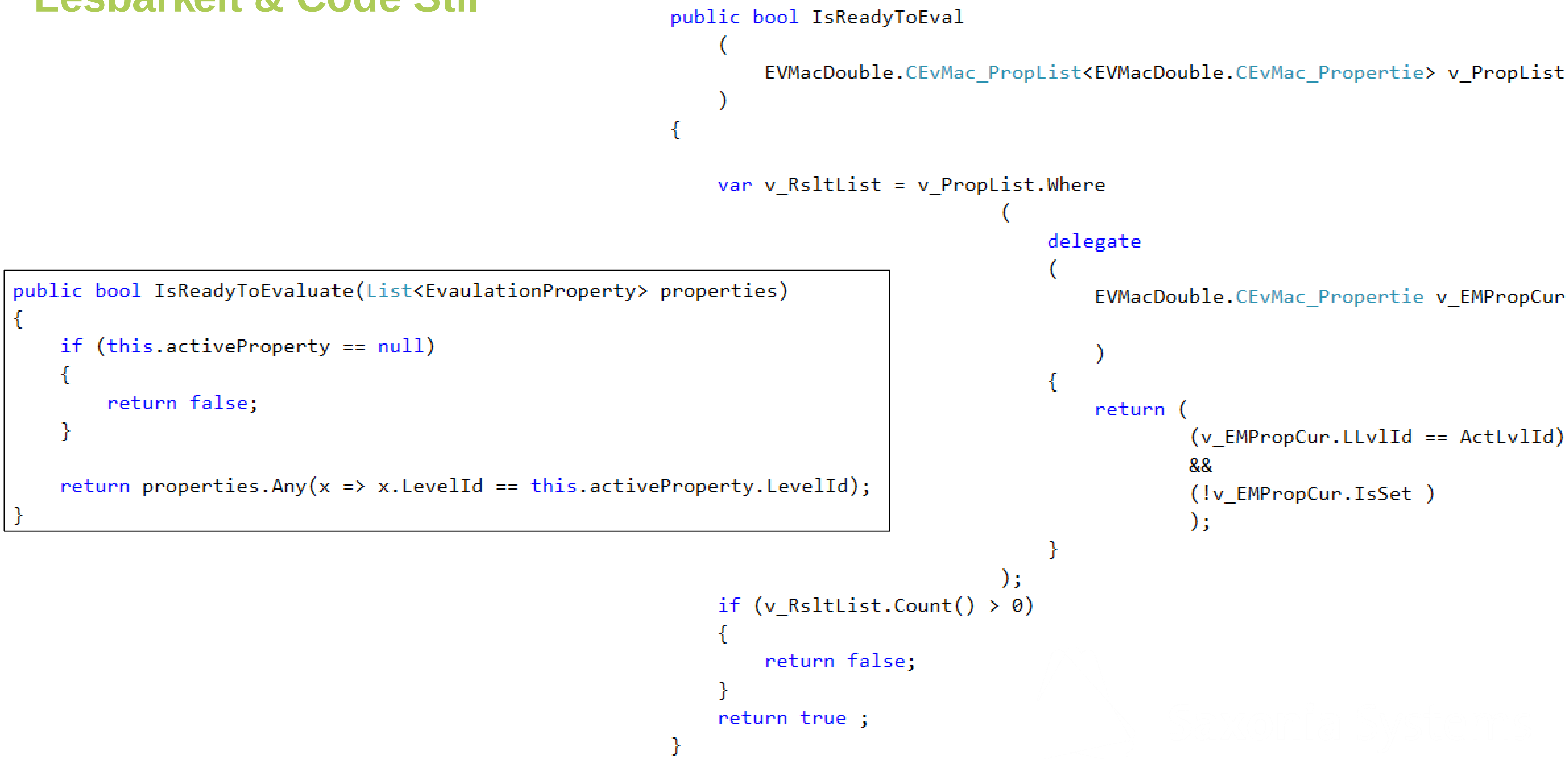
Quelle: Johannes Hofmeister, http://empathic.herokuapp.com

Lesbarkeit & Code Stil



```
- (🟧)⬜:(🟩*)🟫🟨:(🟩*)⬛
{
    🟦= [🟥 🟩:🟫 🟩:⬛];
    if (🟦 {
        // Custom initialization
        if ([🟥 🟩:@🟪(🟩)]) {
          [🟥 🟩:🟪];
        }
       🟦.⬛ = @🟦;

    }
    return 🟦;
}
```

Lesbarkeit & Code Stil



```
public bool IsReadyToEval
    (
        EVMacDouble.CEvMac_PropList<EVMacDouble.CEvMac_Propertie> v_PropList
    )
{

    var v_RsltList = v_PropList.Where
                            (
                                delegate
                                (
                                    EVMacDouble.CEvMac_Propertie v_EMPropCur

                                    )
                                {
                                    return (
                                            (v_EMPropCur.LLvlId == ActLvlId)
                                            &&
                                            (!v_EMPropCur.IsSet )
                                            );
                                }
                            );
    if (v_RsltList.Count() > 0)
    {
        return false;
    }
    return true ;
}
```

```
public bool IsReadyToEvaluate(List<EvaulationProperty> properties)
{
    if (this.activeProperty == null)
    {
        return false;
    }

    return properties.Any(x => x.LevelId == this.activeProperty.LevelId);
}
```

Code Style Guidelines

Statische Analyse von Code mit Einbindung in den Build

Adressiert:

- Dokumentation
- Layout
- Wartbarkeit
- Namensgebung
- Lesbarkeit

{ } StyleCop

checkstyle

R#

JSLint

Religionskriege

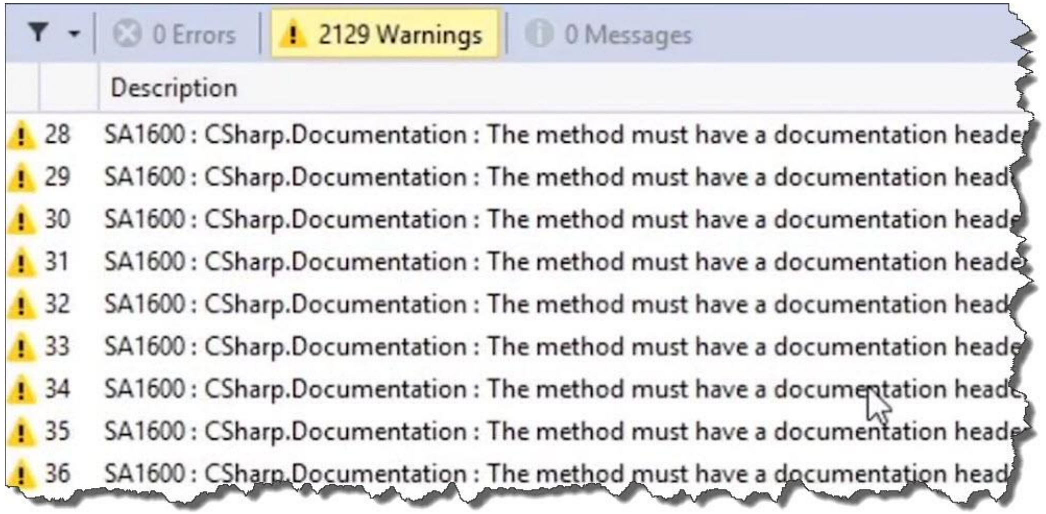
```
private string name;

public string Name
{
    get
    {
        return this.name;
    }

    set
    {
        this.name = value;
    }
}
```

```
private string _name;

public string Name
{
    get { return _name; }

    set { _name = value; }
}
```

```
var name =  "Hendrik";

string name =  "Hendrik";
```

Coding Rules

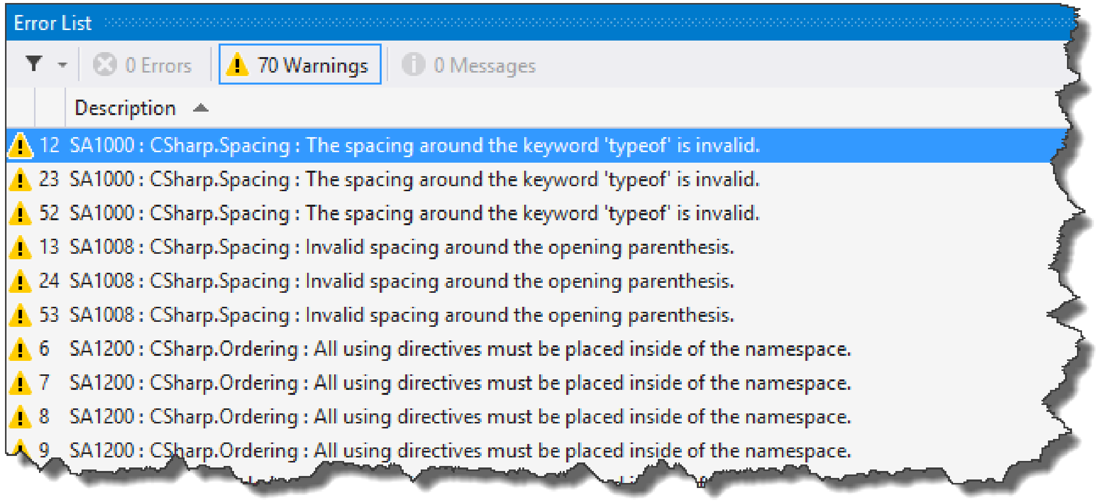
Legacy Software



0 Errors | 2129 Warnings | 0 Messages

		Description
⚠	28	SA1600 : CSharp.Documentation : The method must have a documentation heade
⚠	29	SA1600 : CSharp.Documentation : The method must have a documentation head
⚠	30	SA1600 : CSharp.Documentation : The method must have a documentation heade
⚠	31	SA1600 : CSharp.Documentation : The method must have a documentation heade
⚠	32	SA1600 : CSharp.Documentation : The method must have a documentation head
⚠	33	SA1600 : CSharp.Documentation : The method must have a documentation heade
⚠	34	SA1600 : CSharp.Documentation : The method must have a documentation head
⚠	35	SA1600 : CSharp.Documentation : The method must have a documentation heade
⚠	36	SA1600 : CSharp.Documentation : The method must have a documentation head

Visual Studio Projekttemplate



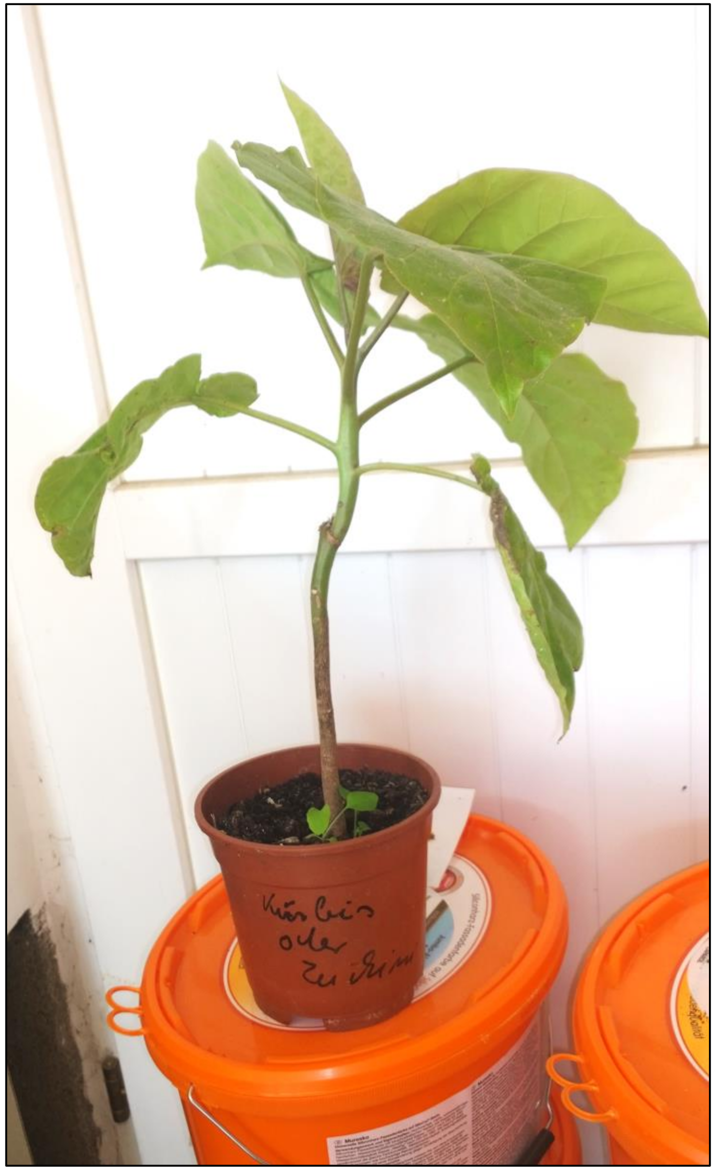
Error List

0 Errors | 70 Warnings | 0 Messages

		Description ▲
⚠	12	SA1000 : CSharp.Spacing : The spacing around the keyword 'typeof' is invalid.
⚠	23	SA1000 : CSharp.Spacing : The spacing around the keyword 'typeof' is invalid.
⚠	52	SA1000 : CSharp.Spacing : The spacing around the keyword 'typeof' is invalid.
⚠	13	SA1008 : CSharp.Spacing : Invalid spacing around the opening parenthesis.
⚠	24	SA1008 : CSharp.Spacing : Invalid spacing around the opening parenthesis.
⚠	53	SA1008 : CSharp.Spacing : Invalid spacing around the opening parenthesis.
⚠	6	SA1200 : CSharp.Ordering : All using directives must be placed inside of the namespace.
⚠	7	SA1200 : CSharp.Ordering : All using directives must be placed inside of the namespace.
⚠	8	SA1200 : CSharp.Ordering : All using directives must be placed inside of the namespace.
⚠	9	SA1200 : CSharp.Ordering : All using directives must be placed inside of the namespace.

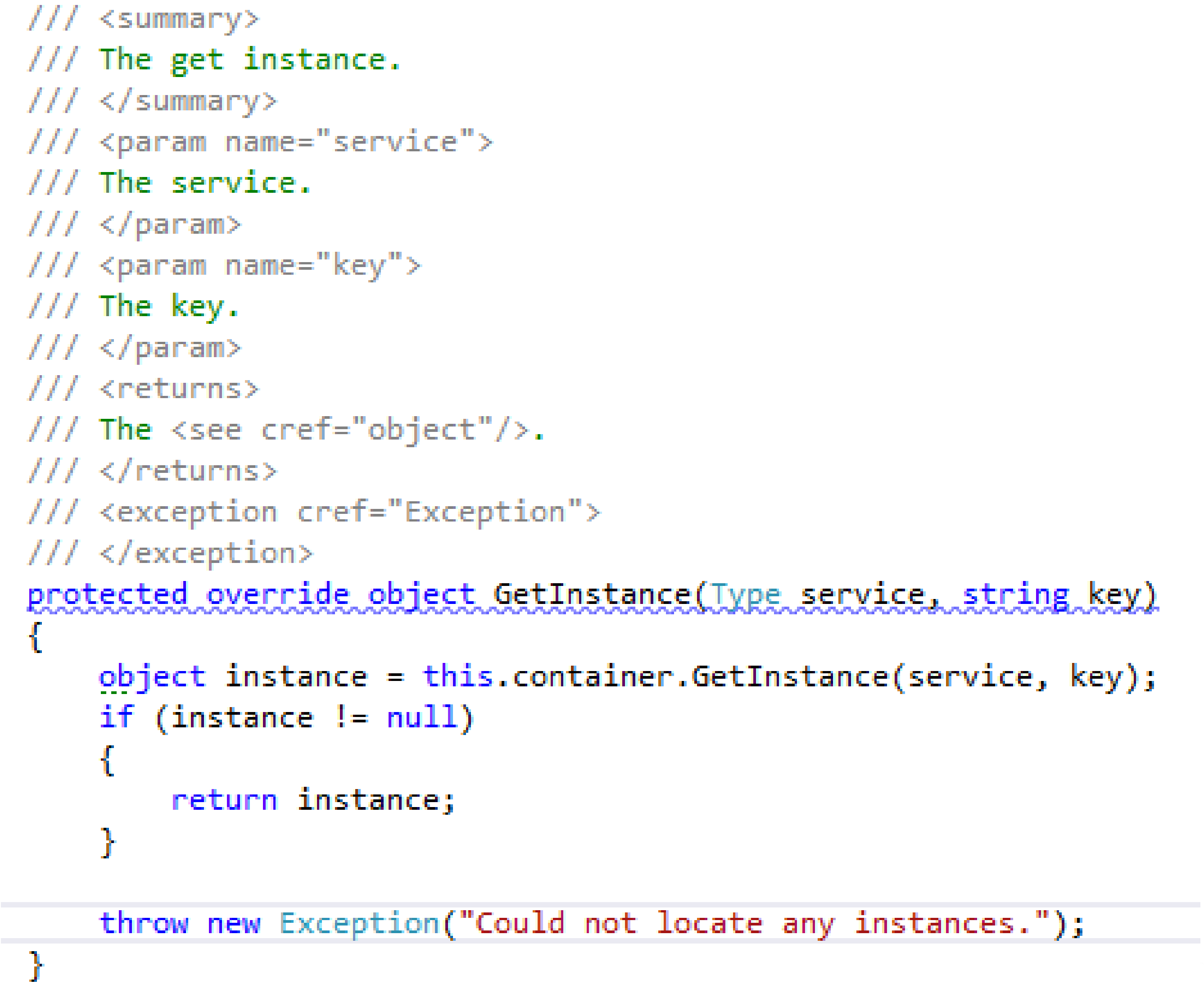
Kommentare und Bezeichnungen

Kommentare



Kommentare

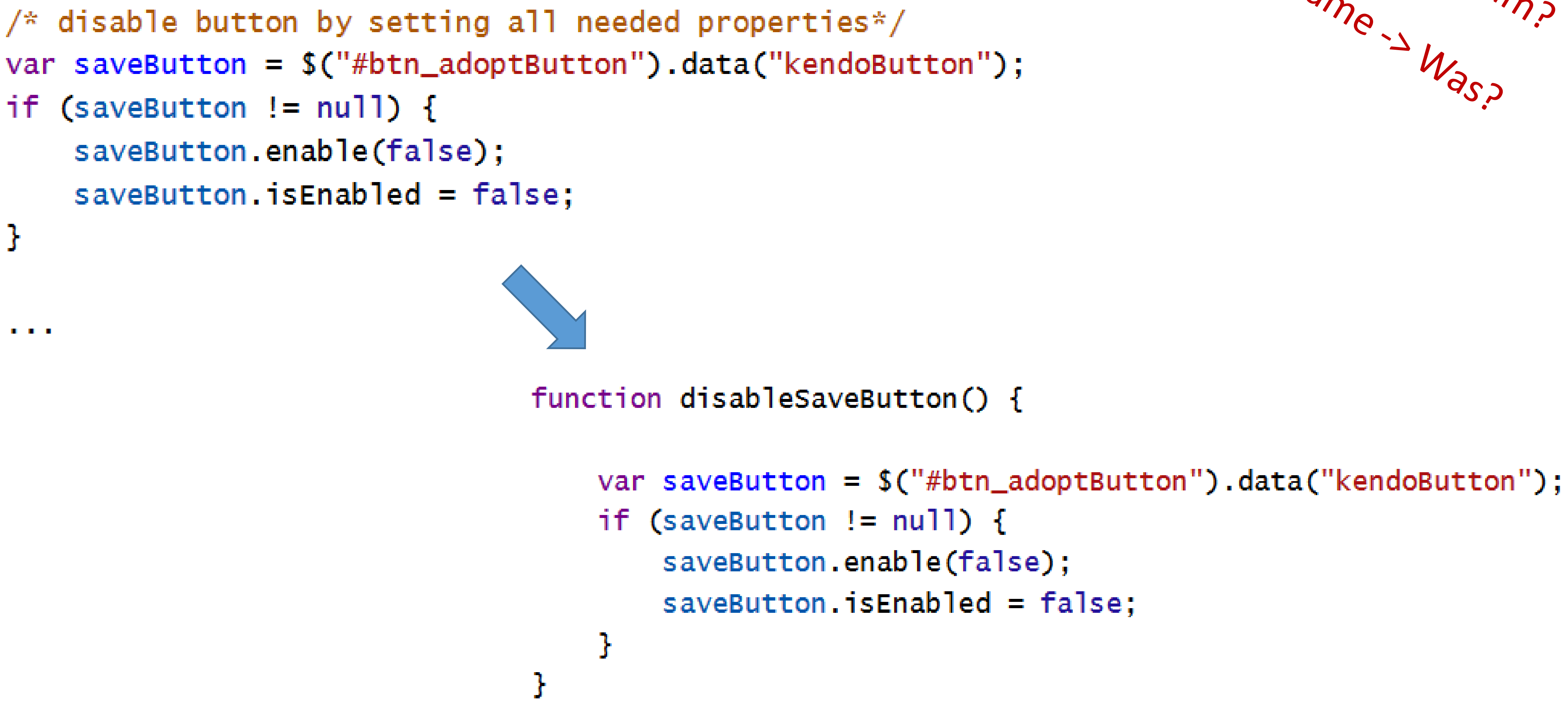
Code -> Wie?
Kommentar -> Warum?



```
/// <summary>
/// The get instance.
/// </summary>
/// <param name="service">
/// The service.
/// </param>
/// <param name="key">
/// The key.
/// </param>
/// <returns>
/// The <see cref="object"/>.
/// </returns>
/// <exception cref="Exception">
/// </exception>
protected override object GetInstance(Type service, string key)
{
    object instance = this.container.GetInstance(service, key);
    if (instance != null)
    {
        return instance;
    }

    throw new Exception("Could not locate any instances.");
}
```

Kommentare

Code -> Wie?
Kommentar -> Warum?
Methodenname -> Was?



```
/* disable button by setting all needed properties*/
var saveButton = $("#btn_adoptButton").data("kendoButton");
if (saveButton != null) {
    saveButton.enable(false);
    saveButton.isEnabled = false;
}

...
```

```
function disableSaveButton() {

    var saveButton = $("#btn_adoptButton").data("kendoButton");
    if (saveButton != null) {
        saveButton.enable(false);
        saveButton.isEnabled = false;
    }
}
```

Methoden statt Kommentare

```
public IContainer CreateDependencyContainer()
{
    var builder = new ContainerBuilder();

    // Register modules
    builder.RegisterModule<InMemoryModule>();
    builder.RegisterModule<SapStoreModule>();

    // Register infrastructure
    builder.RegisterType<BingMapService>().As<IMapService>().SingleInstance();
    builder.RegisterType<SettingsHandler>().As<ISettingsHandler>().SingleInstance();
    builder.RegisterType<UserAuthorization>().As<IUserAuthorization>();

    // Page View Models
    builder.RegisterType<LoginPageViewModel>().As<LoginPageViewModel>().SingleInstance();
    builder.RegisterType<RouteSelectionPageViewModel>().As<RouteSelectionPageViewModel>().SingleInstance();
    builder.RegisterType<RoutePageViewModel>().As<RoutePageViewModel>().SingleInstance();
    builder.RegisterType<MapViewPageViewModel>().As<MapViewPageViewModel>().SingleInstance();
    builder.RegisterType<ConnectionObjectPageViewModel>().As<ConnectionObjectPageViewModel>().SingleInstance();
    builder.RegisterType<ReadingPageViewModel>().As<ReadingPageViewModel>().SingleInstance();

    // Settings View Models
    builder.RegisterType<GeneralSettingsFlyoutViewModel>().As<GeneralSettingsFlyoutViewModel>();

    // Map ViewModel
    builder.RegisterType<MapServiceControlViewModel>().As<MapServiceControlViewModel>().SingleInstance();

    return builder.Build();
}
```

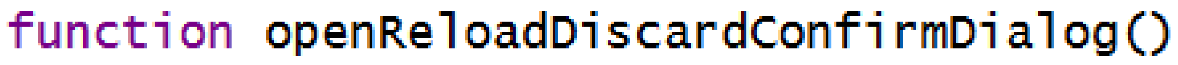
Methoden statt Kommentare

```
public IContainer CreateDependencyContainer()
{
    var builder = new ContainerBuilder();

    RegisterModules(builder);
    RegisterInfrastructure(builder);
    RegisterViewModels(builder);

    return builder.Build();
}

private static void RegisterModules(ContainerBuilder builder)
{
     …
}

private static void RegisterInfrastructure(ContainerBuilder builder)
{
    …
}

private static void RegisterViewModels(ContainerBuilder builder)
{
    …
}
```

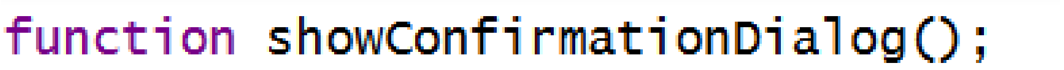
Methodennamen

Principle of Least Astonishment???

to open, reload, discard and confirm dialog…



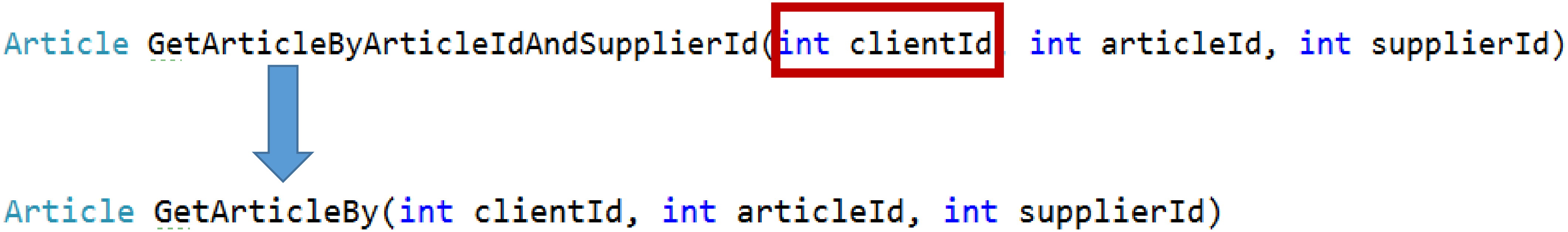
```
function openReloadDiscardConfirmDialog()
```



```
function showConfirmationDialog();
```

Methodennamen

Information Hiding!!!



```
Article GetArticleByArticleIdAndSupplierId(int clientId, int articleId, int supplierId)
```

```
Article GetArticleBy(int clientId, int articleId, int supplierId)
```

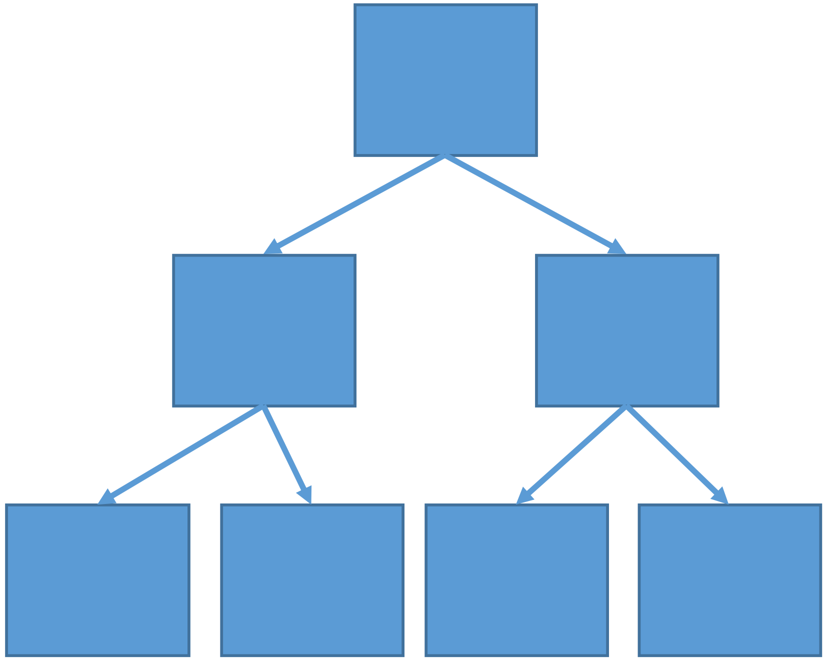
Abhängigkeiten

Was ist schlechtes Design?!?

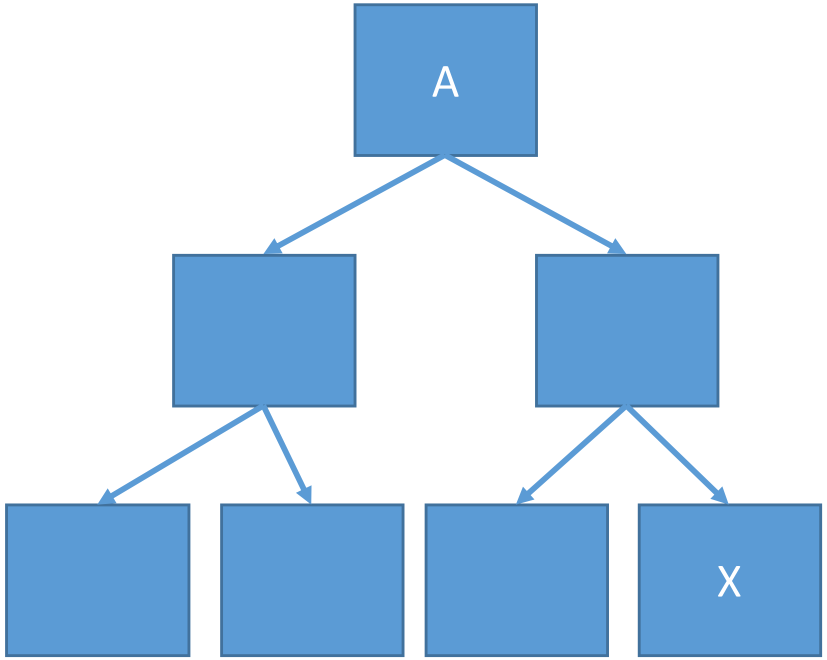
1. It is hard to change because every change affects too many other parts of the system. (Rigidity - Starr)
2. When you make a change, unexpected parts of the system break. (Fragility - Zerbrechlich)
3. It is hard to reuse in another application because it cannot be disentangled from the current application. (Immobility - Unbeweglich)

Quelle: http://www.objectmentor.com/resources/articles/dip.pdf

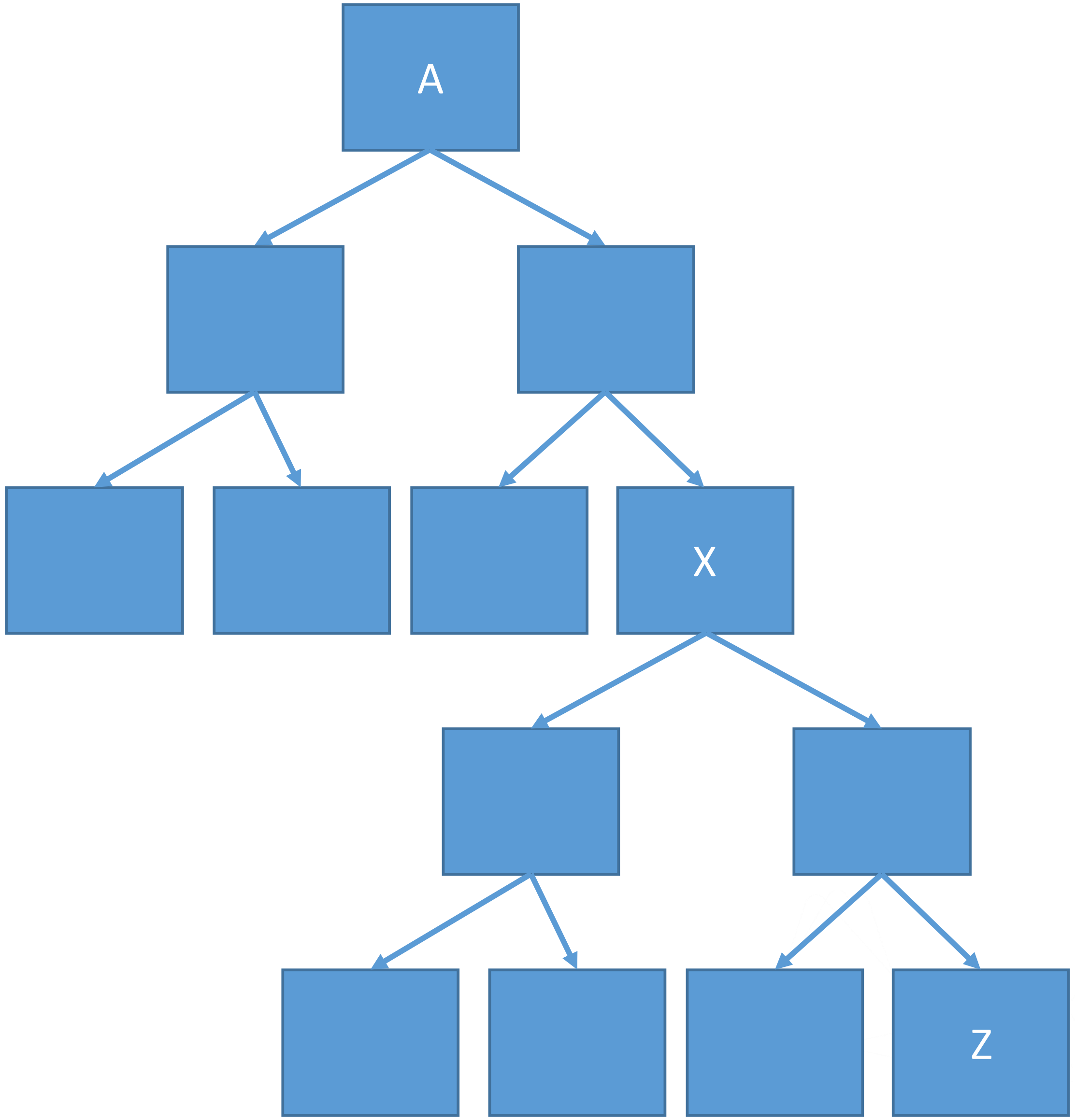
Abhängigkeiten



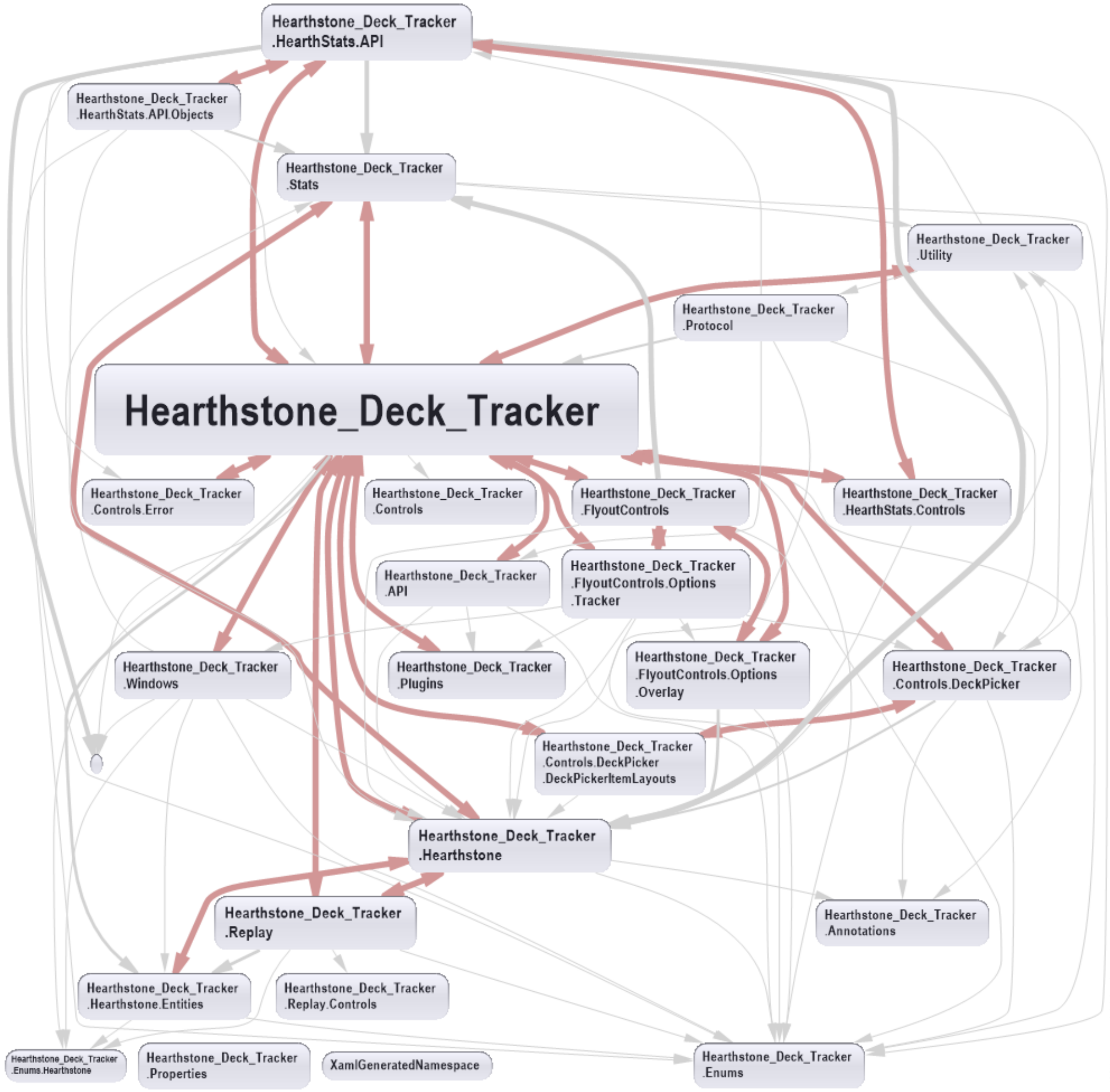
Abhängigkeiten



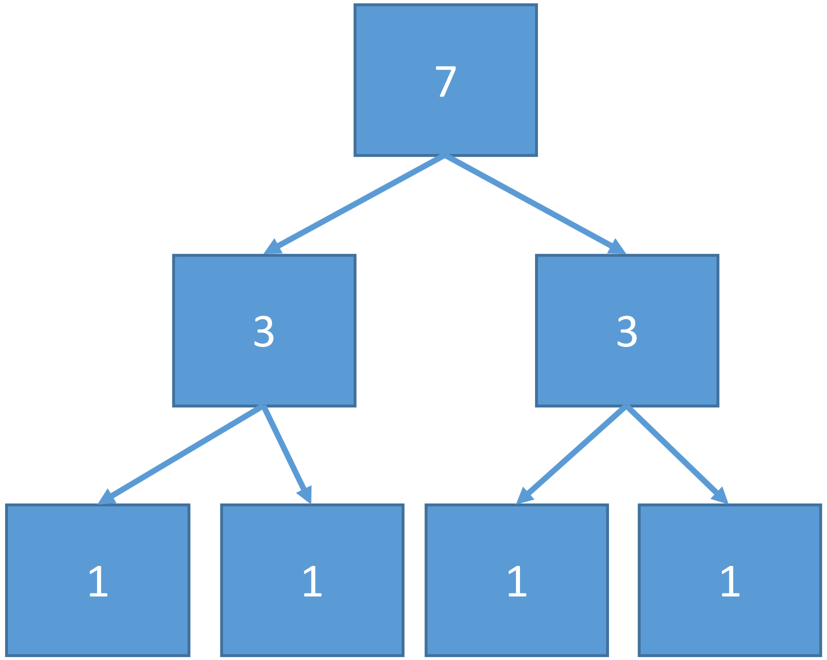
Abhängigkeiten



Abhängigkeiten


Hearthstone_Deck_Tracker .HearthStats.API
Hearthstone_Deck_Tracker .HearthStats.API.Objects
Hearthstone_Deck_Tracker .Stats
Hearthstone_Deck_Tracker .Utility
Hearthstone_Deck_Tracker .Protocol
Hearthstone_Deck_Tracker
Hearthstone_Deck_Tracker .Controls.Error
Hearthstone_Deck_Tracker .Controls
Hearthstone_Deck_Tracker .FlyoutControls
Hearthstone_Deck_Tracker .HearthStats.Controls
Hearthstone_Deck_Tracker .API
Hearthstone_Deck_Tracker .FlyoutControls.Options .Tracker
Hearthstone_Deck_Tracker .Windows
Hearthstone_Deck_Tracker .Plugins
Hearthstone_Deck_Tracker .FlyoutControls.Options .Overlay
Hearthstone_Deck_Tracker .Controls.DeckPicker
Hearthstone_Deck_Tracker .Controls.DeckPicker .DeckPickerItemLayouts
Hearthstone_Deck_Tracker .Hearthstone
Hearthstone_Deck_Tracker .Replay
Hearthstone_Deck_Tracker .Annotations
Hearthstone_Deck_Tracker .Hearthstone.Entities
Hearthstone_Deck_Tracker .Replay.Controls
Hearthstone_Deck_Tracker .Enums.Hearthstone
Hearthstone_Deck_Tracker .Properties
XamlGeneratedNamespace
Hearthstone_Deck_Tracker .Enums

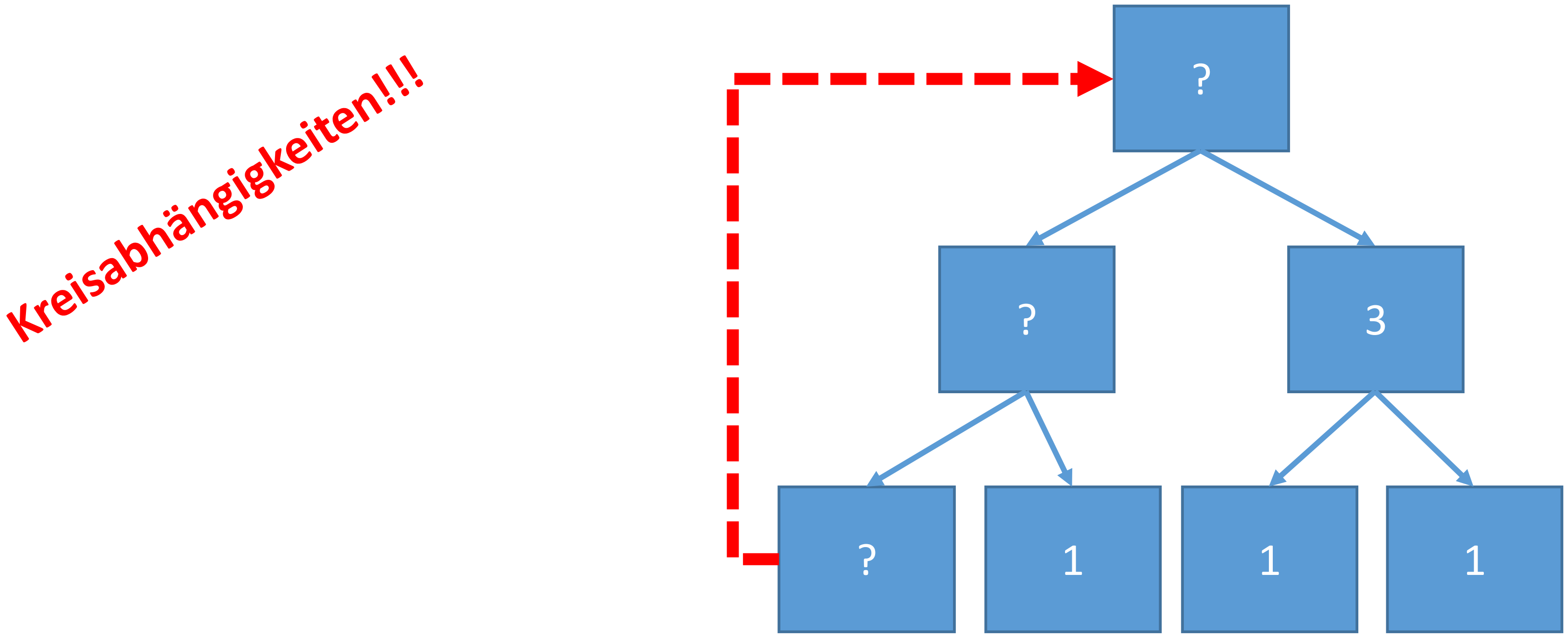
Abhängigkeiten



Kumulierte Komponentenabhängigkeit = 1*7 + 2*3 + 4*1 = 17

(cumulated component dependency)

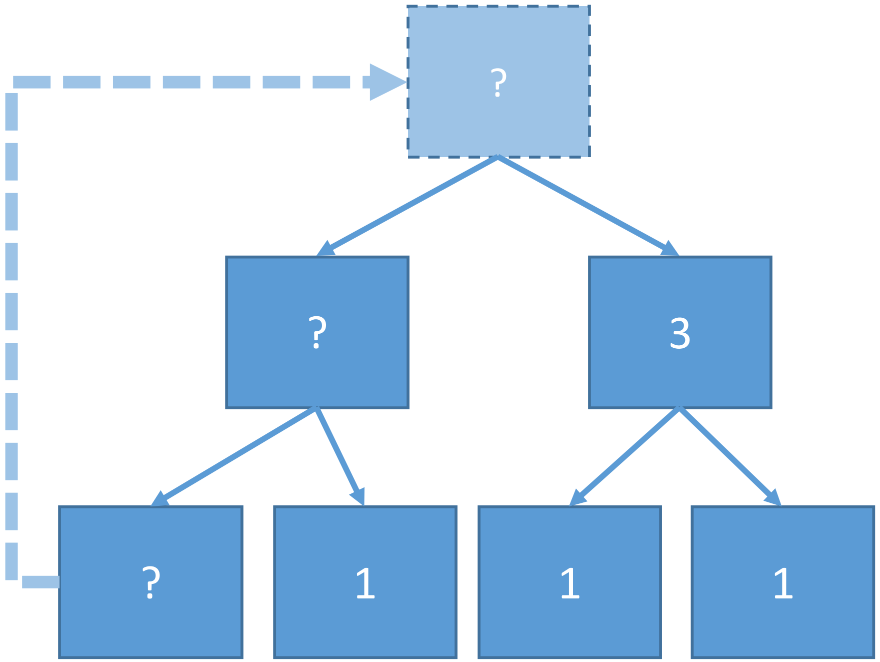
Abhängigkeiten



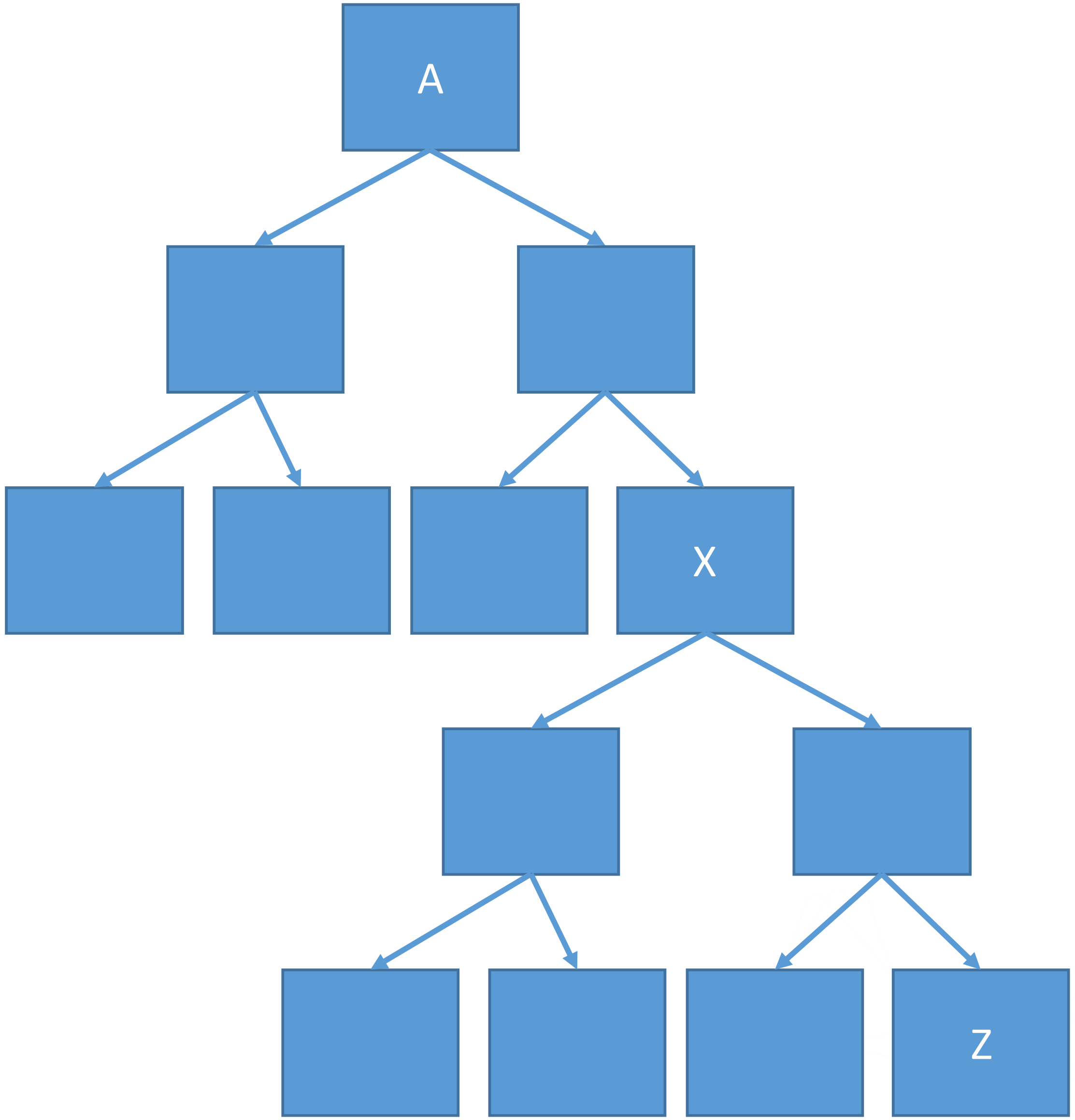
Kumulierte Komponentenabhängigkeit = 7^2 = 49

(cumulated component dependency)

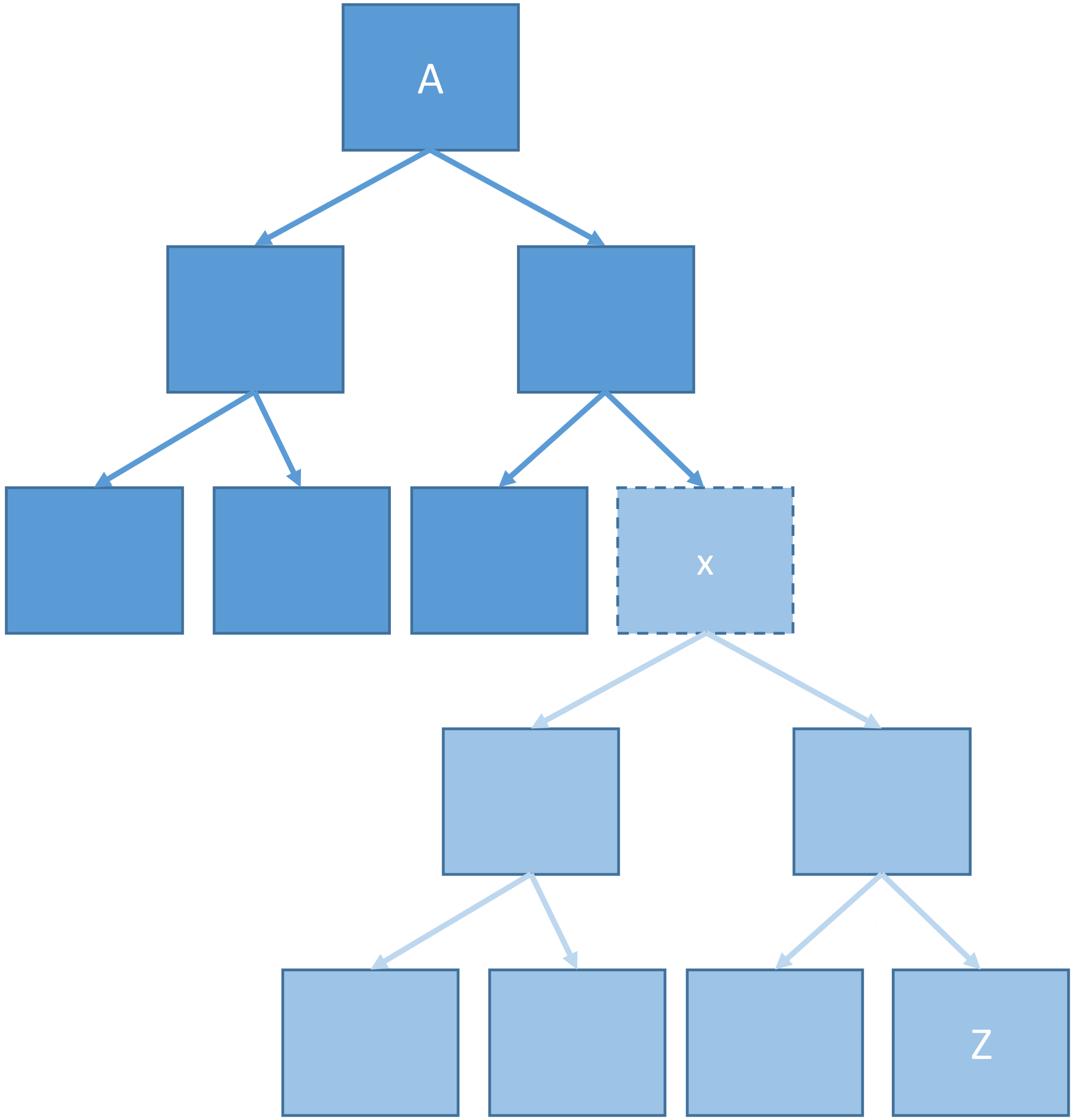
Abhängigkeiten



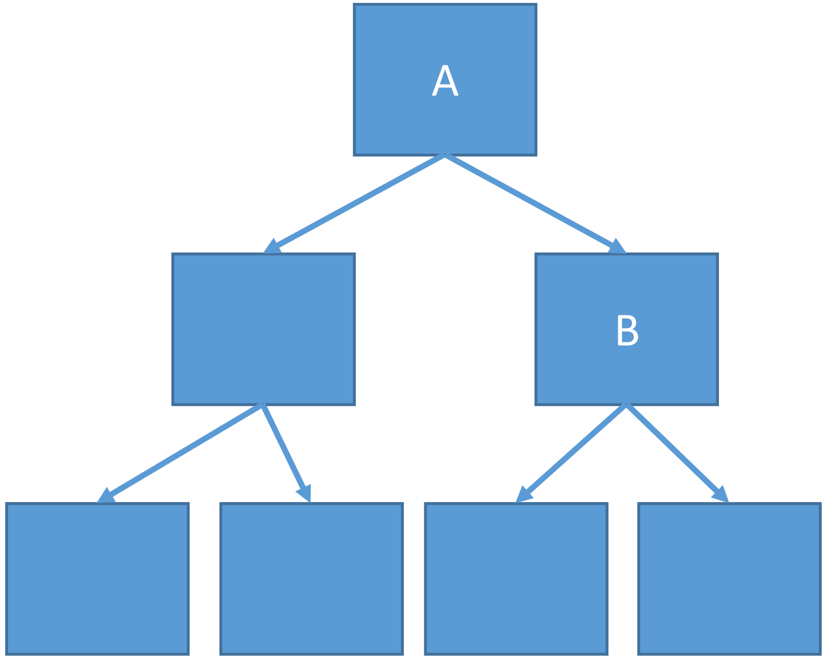
Abhängigkeiten



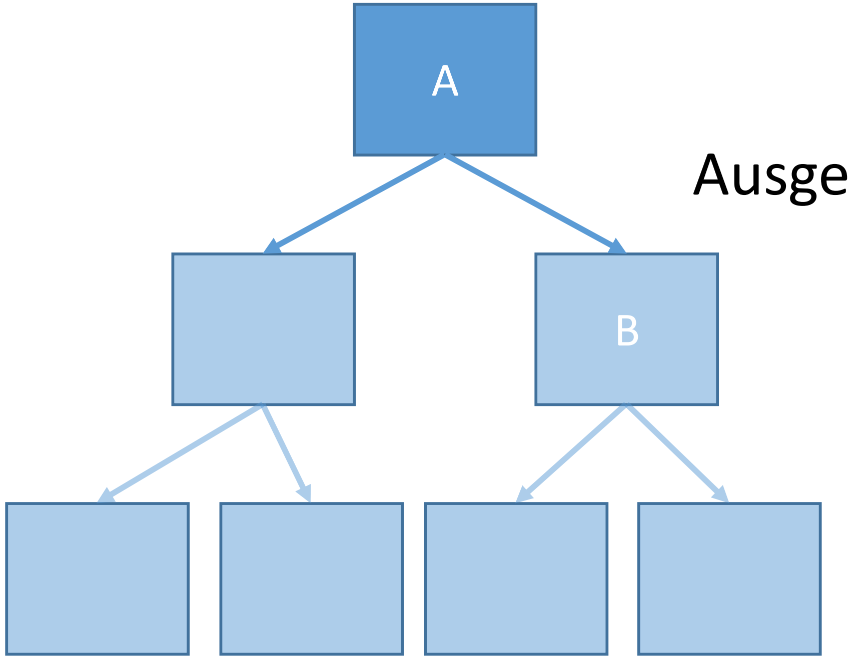
Abhängigkeiten



Abhängigkeiten

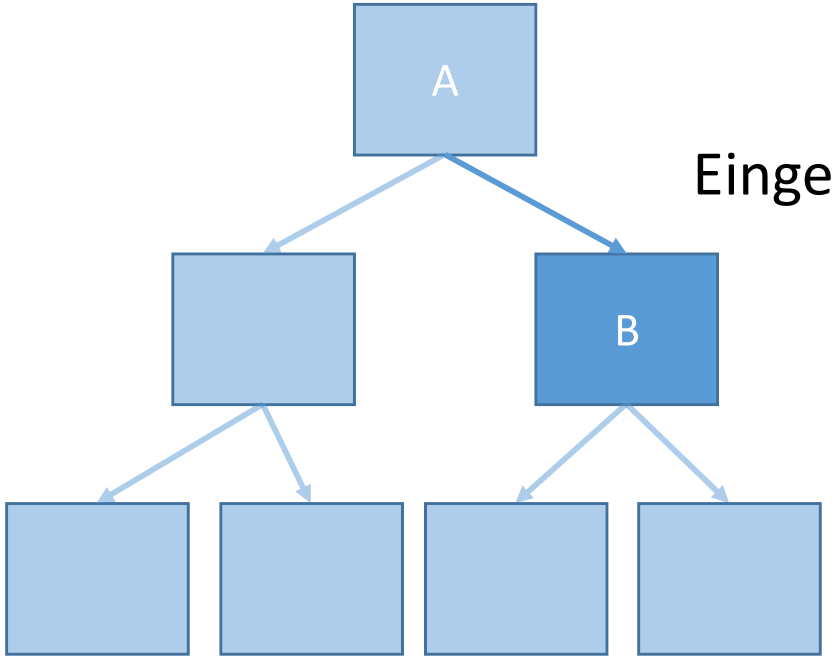


Abhängigkeiten



Ausgehende Abhängigkeiten = 2

Abhängigkeiten



Eingehende Abhängigkeiten = 1

Instability vs. Abstractness

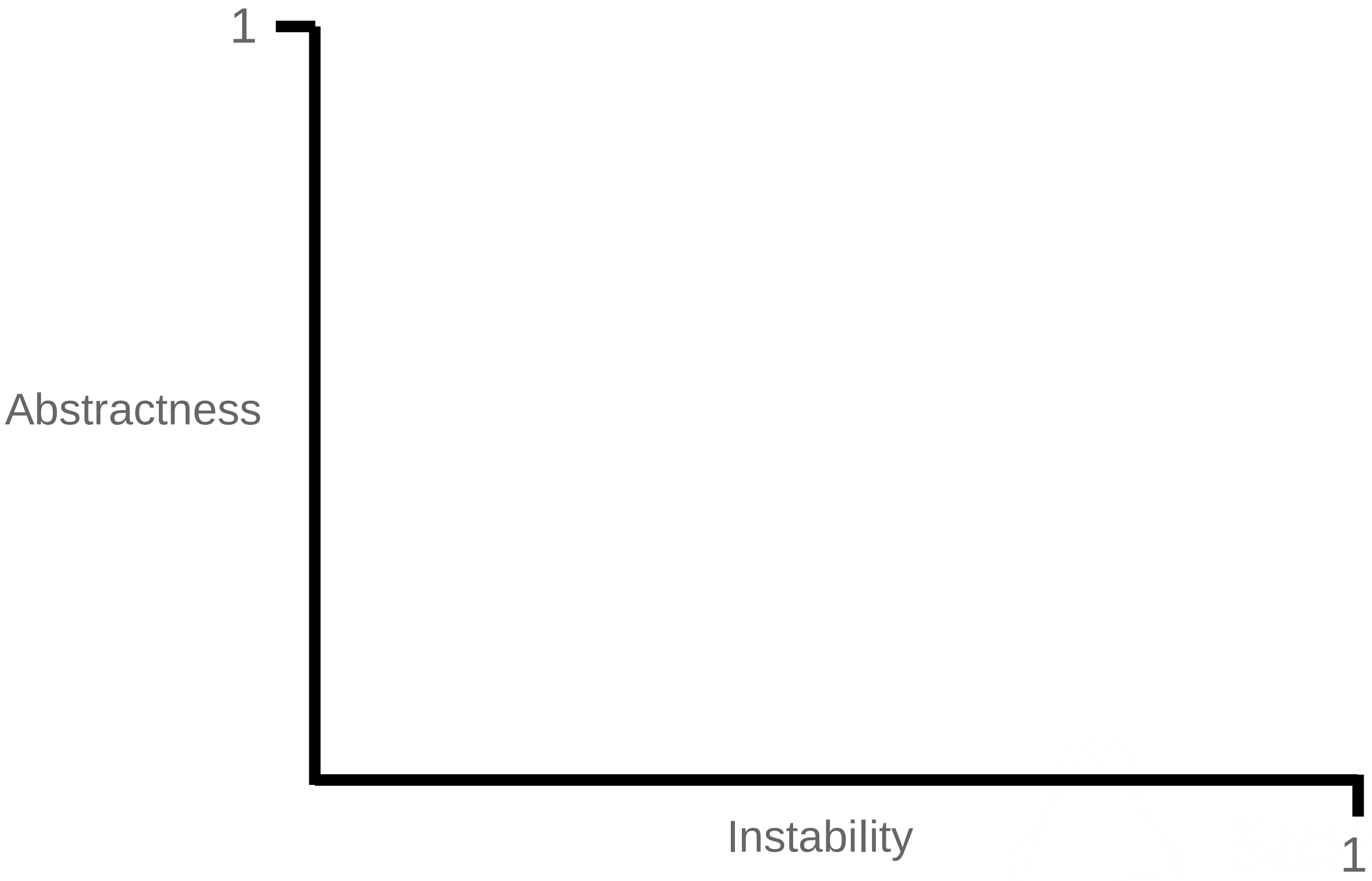
$$\text{Instability} = Ce / (Ce + Ca)$$

$$\text{Abstractness} = Na / N$$

Ce	– Efferent Coupling (Ausgehende Abhängigkeiten)
Ca	– Afferent Coupling (Eingehende Abhängigkeiten)
N	– Anzahl Typen
Na	– Anzahl abstrakte Typen

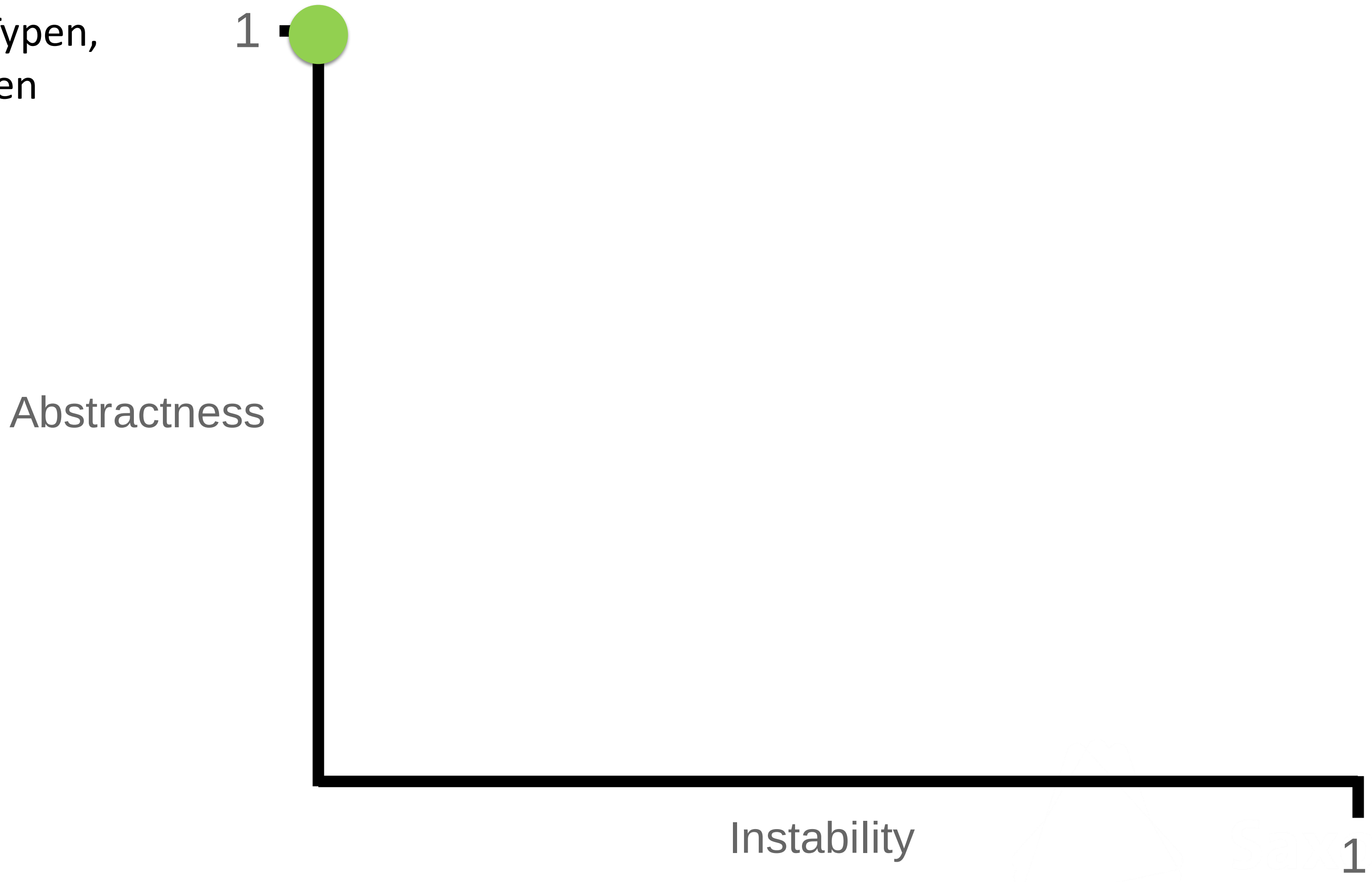
Quelle: http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod

Instability vs. Abstractness



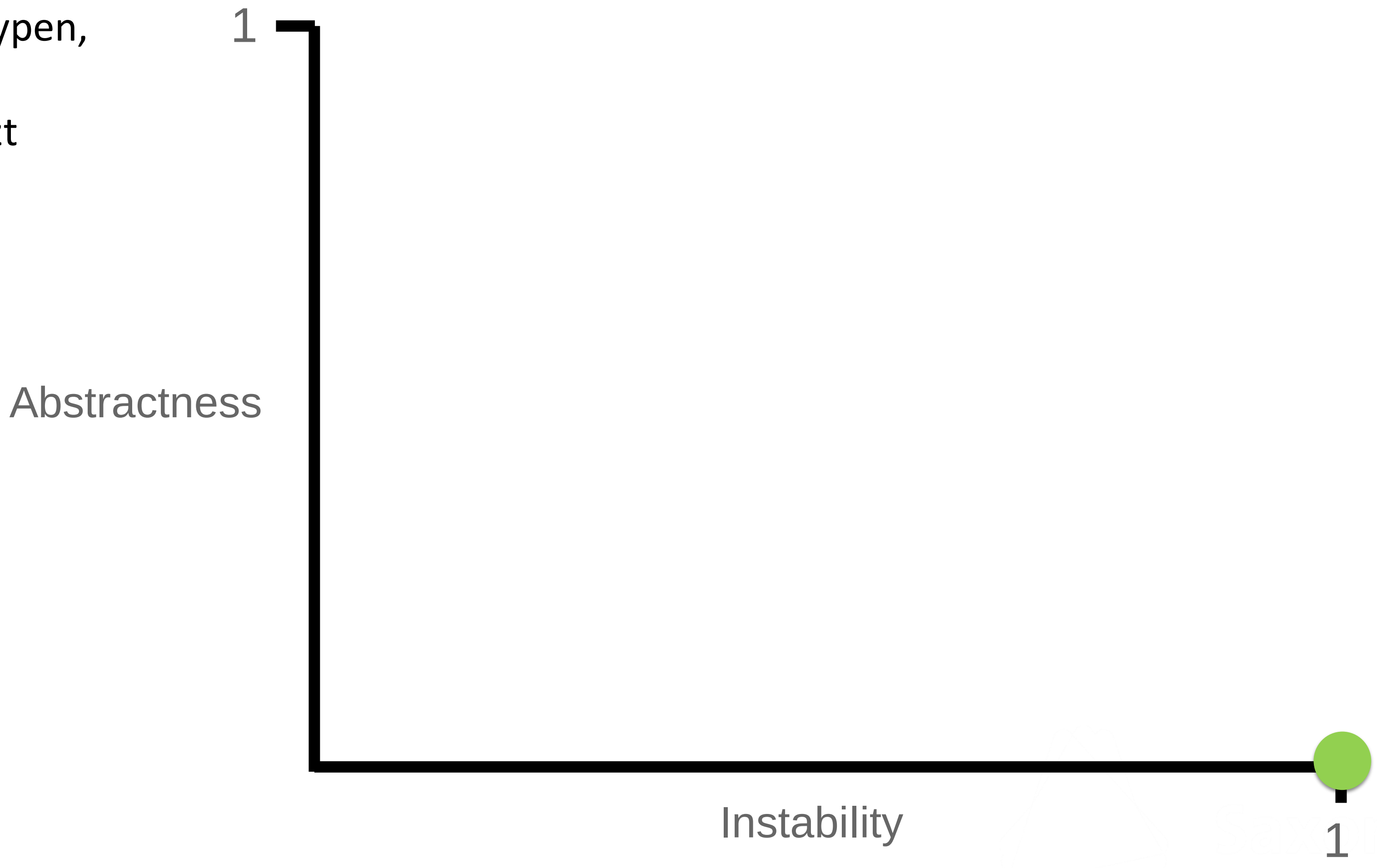
Instability vs. Abstractness

Es gibt nur abstrakte Typen, die selbst keine anderen Typen nutzen.



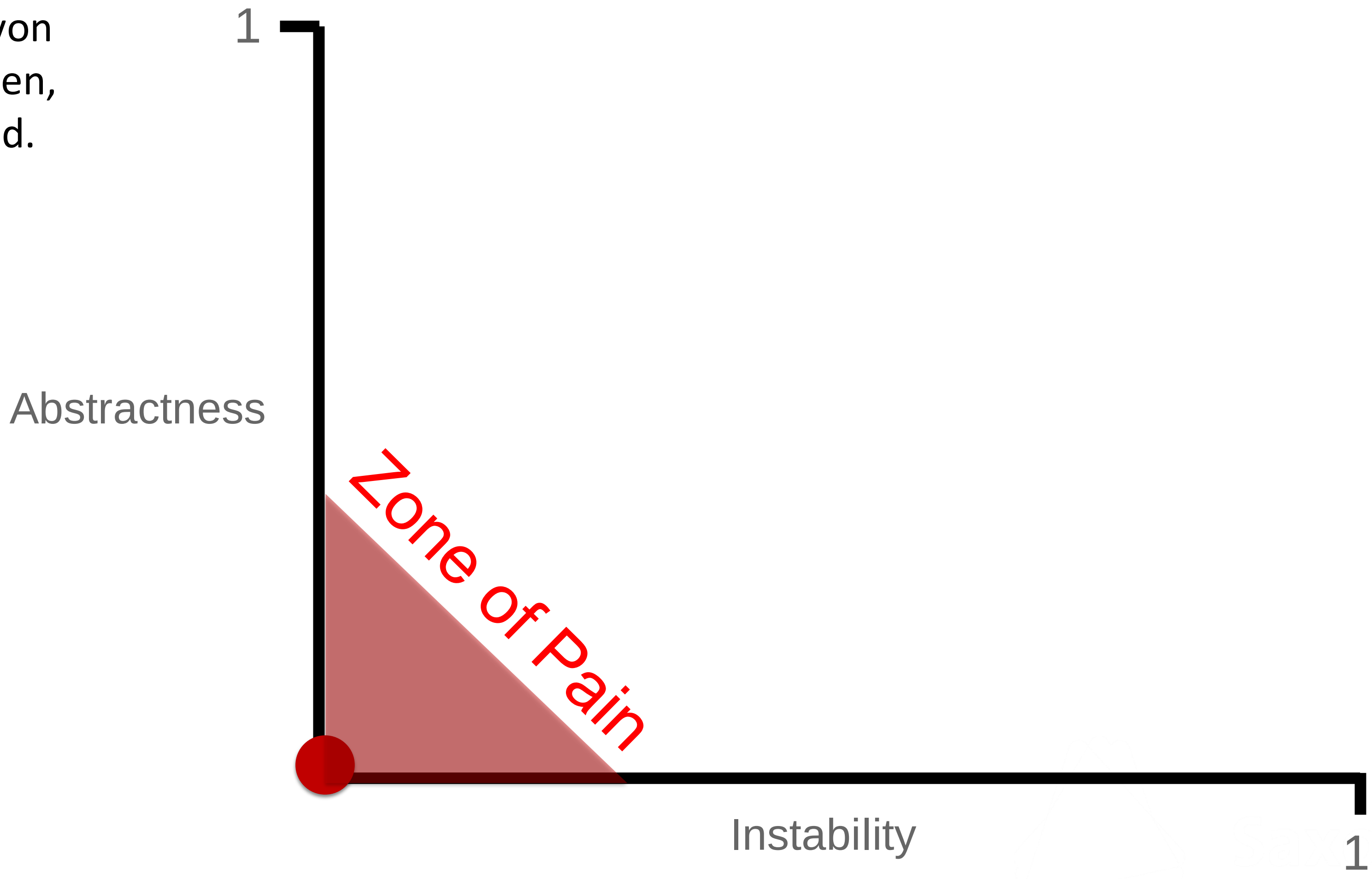
Instability vs. Abstractness

Es gibt nur konkrete Typen, die selbst nicht von anderen Typen genutzt werden.



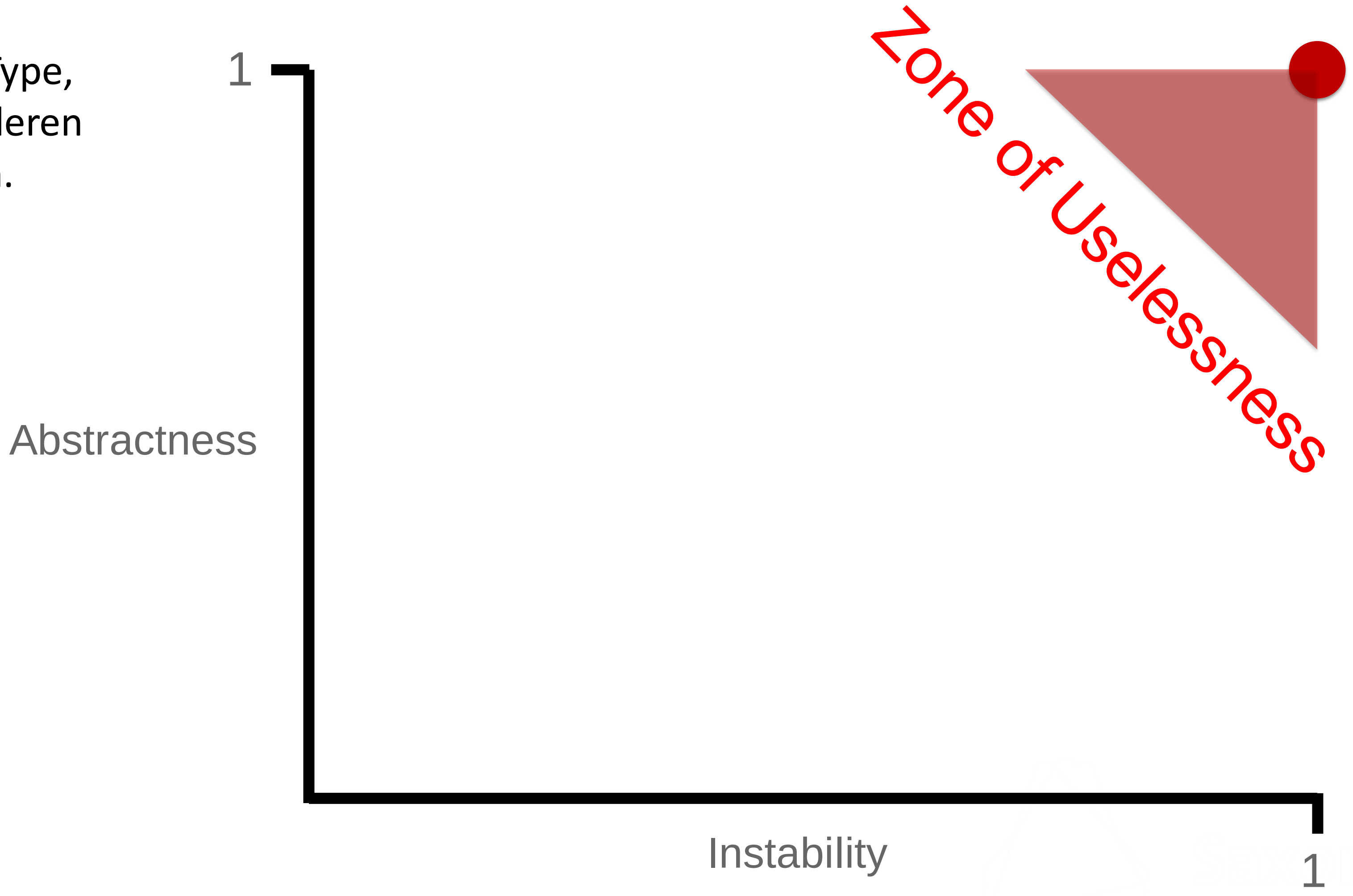
Instability vs. Abstractness

Es gibt nur Typen die von anderen genutzt werden, aber nicht abstrakt sind.



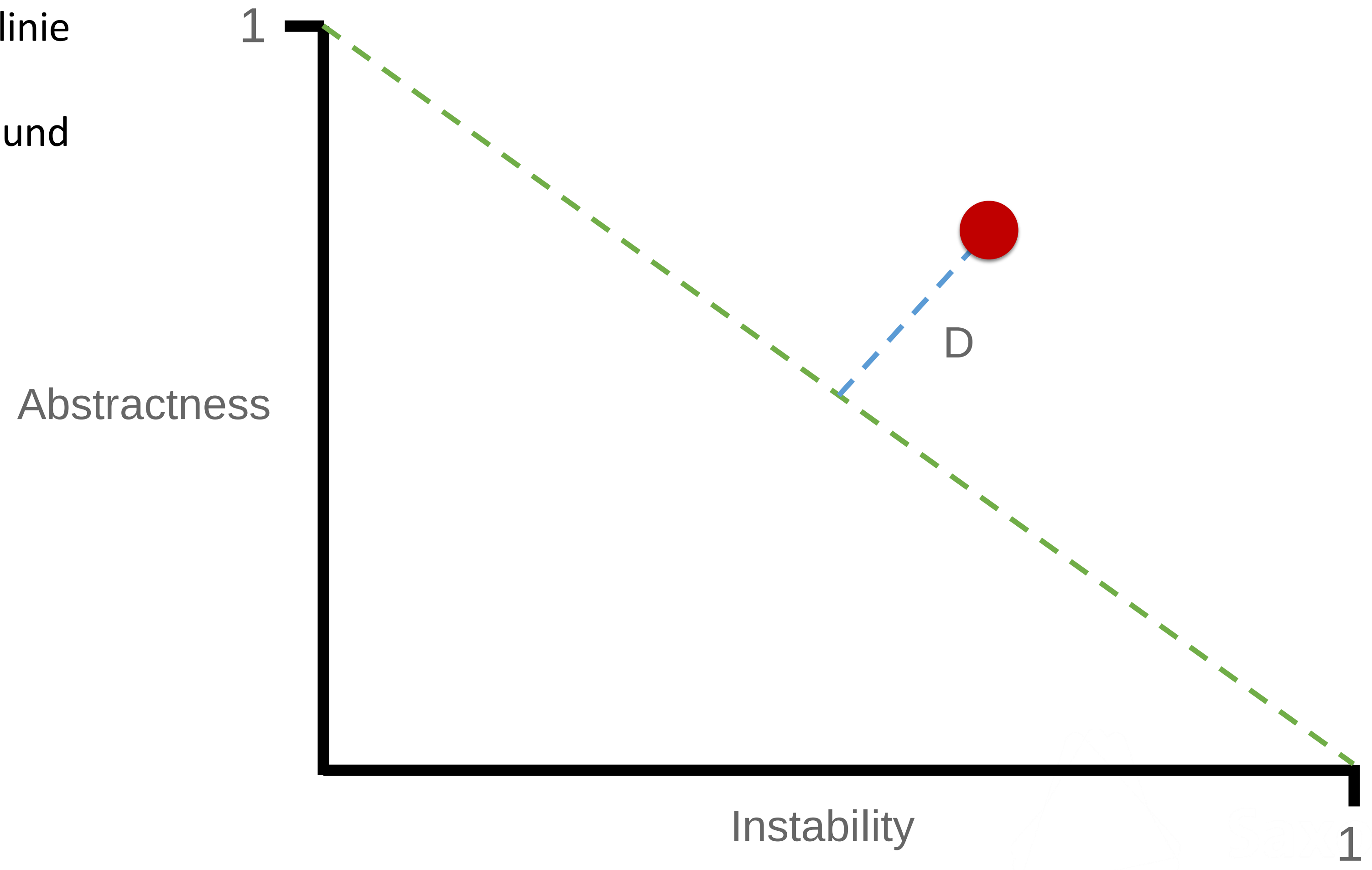
Instability vs. Abstractness

Es gibt nur abstrakte Type, die aber nicht von anderen Typen genutzt werden.

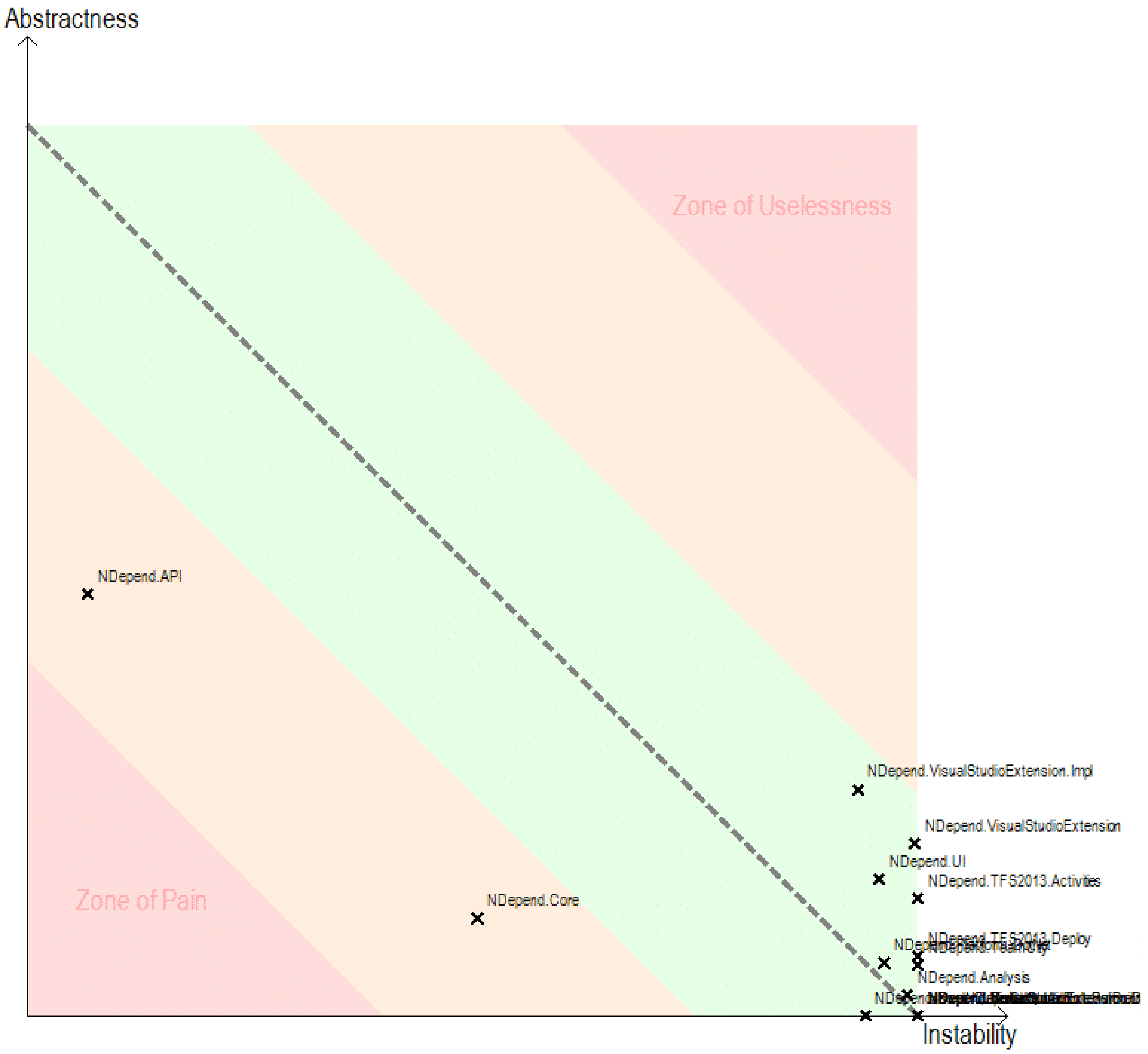


Instability vs. Abstractness

Der Abstand zur Ideallinie stellt das Verhältnis zwischen Abstraktion und Instabilität dar.

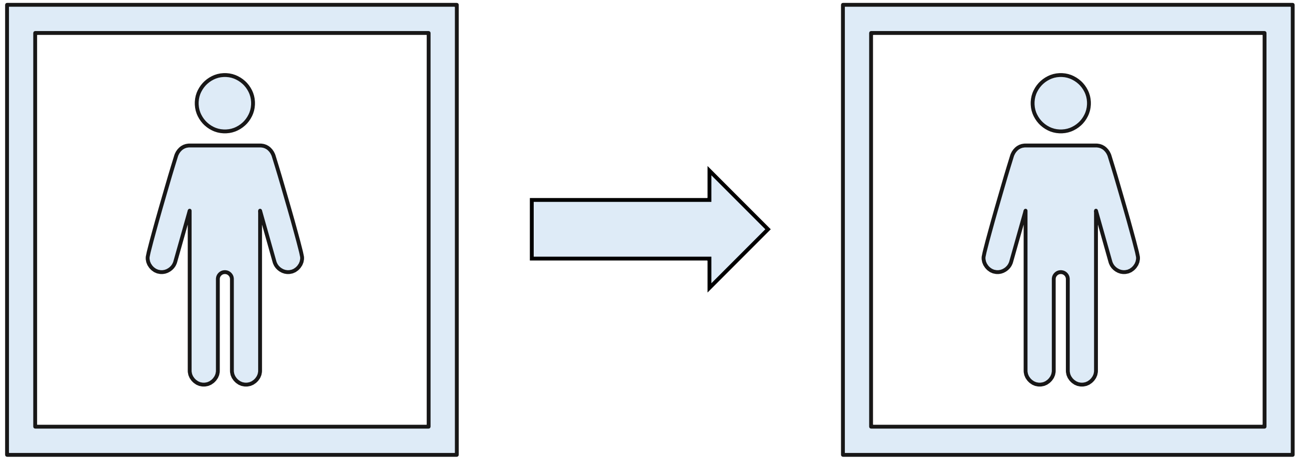


Instability vs. Abstractness

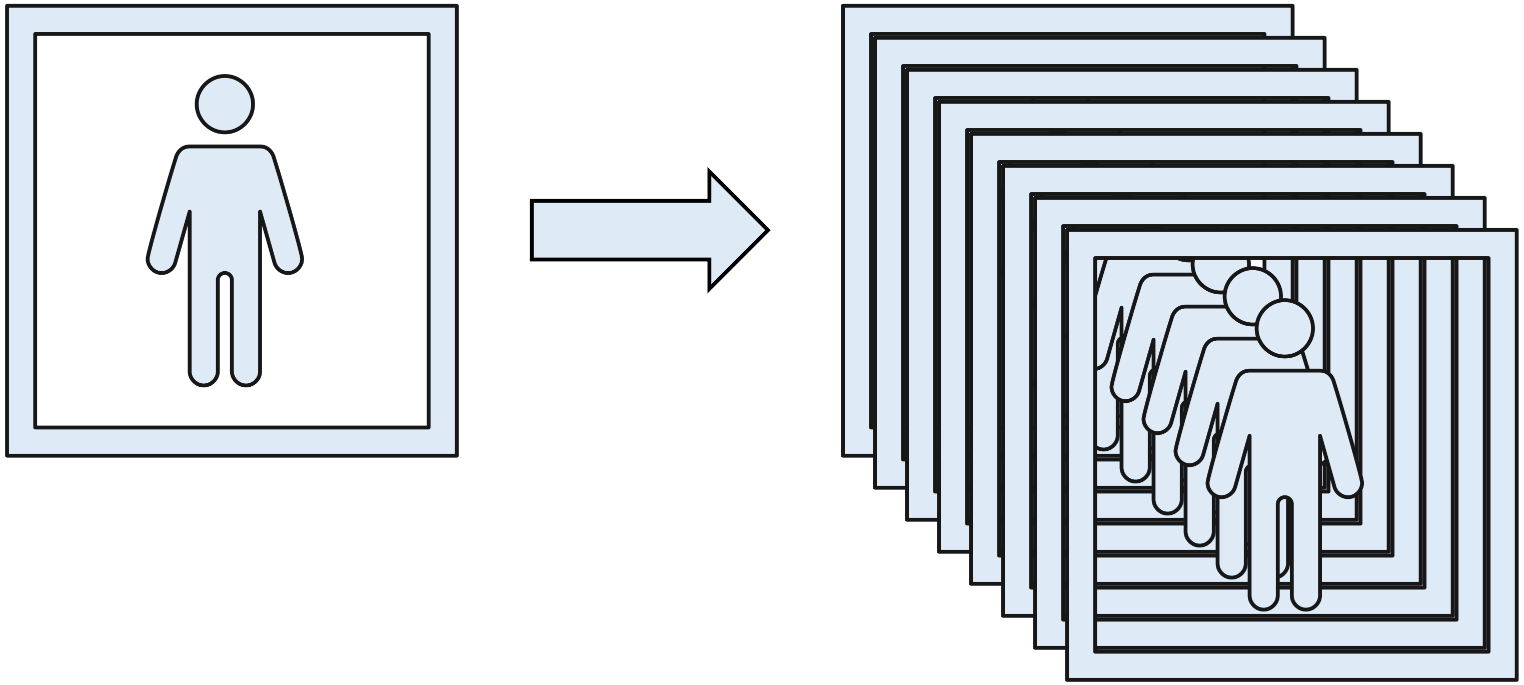


Code Clones

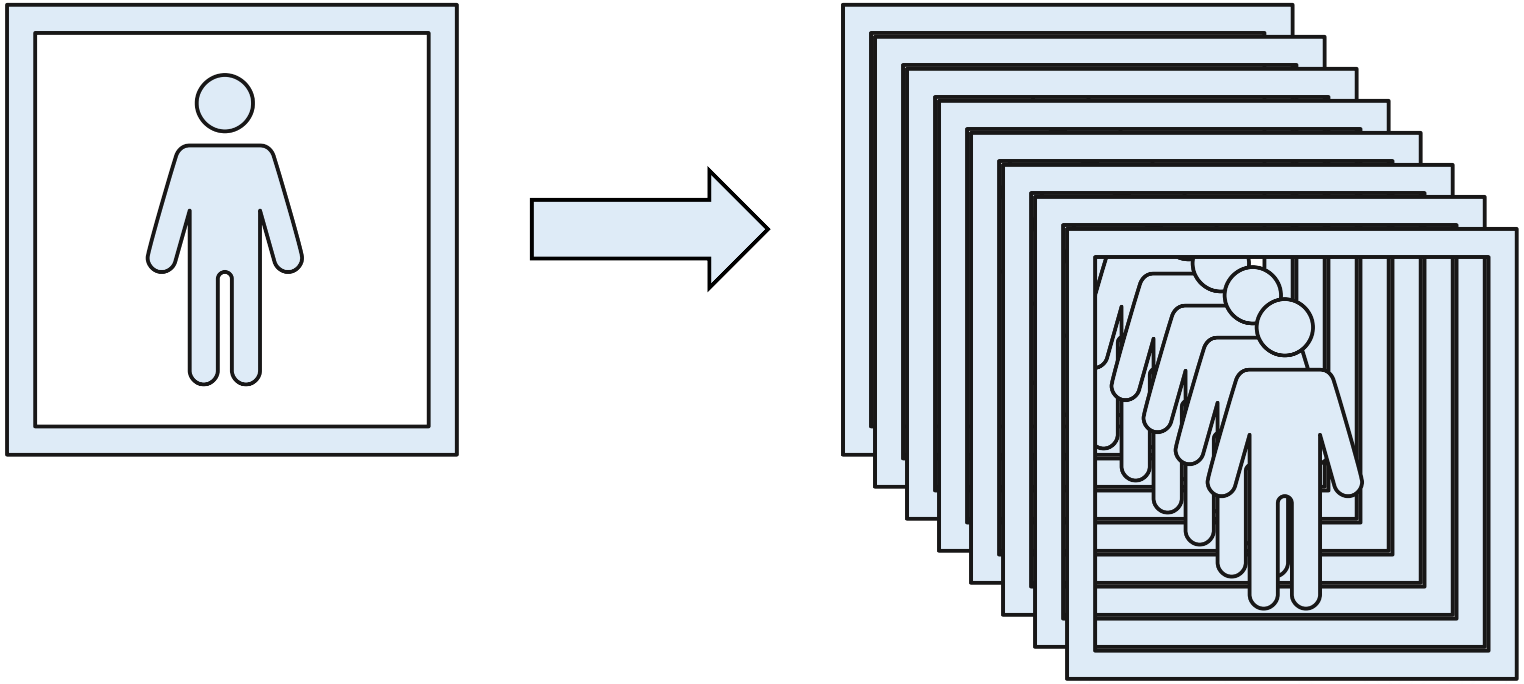
Code Clones vs. Duplikate



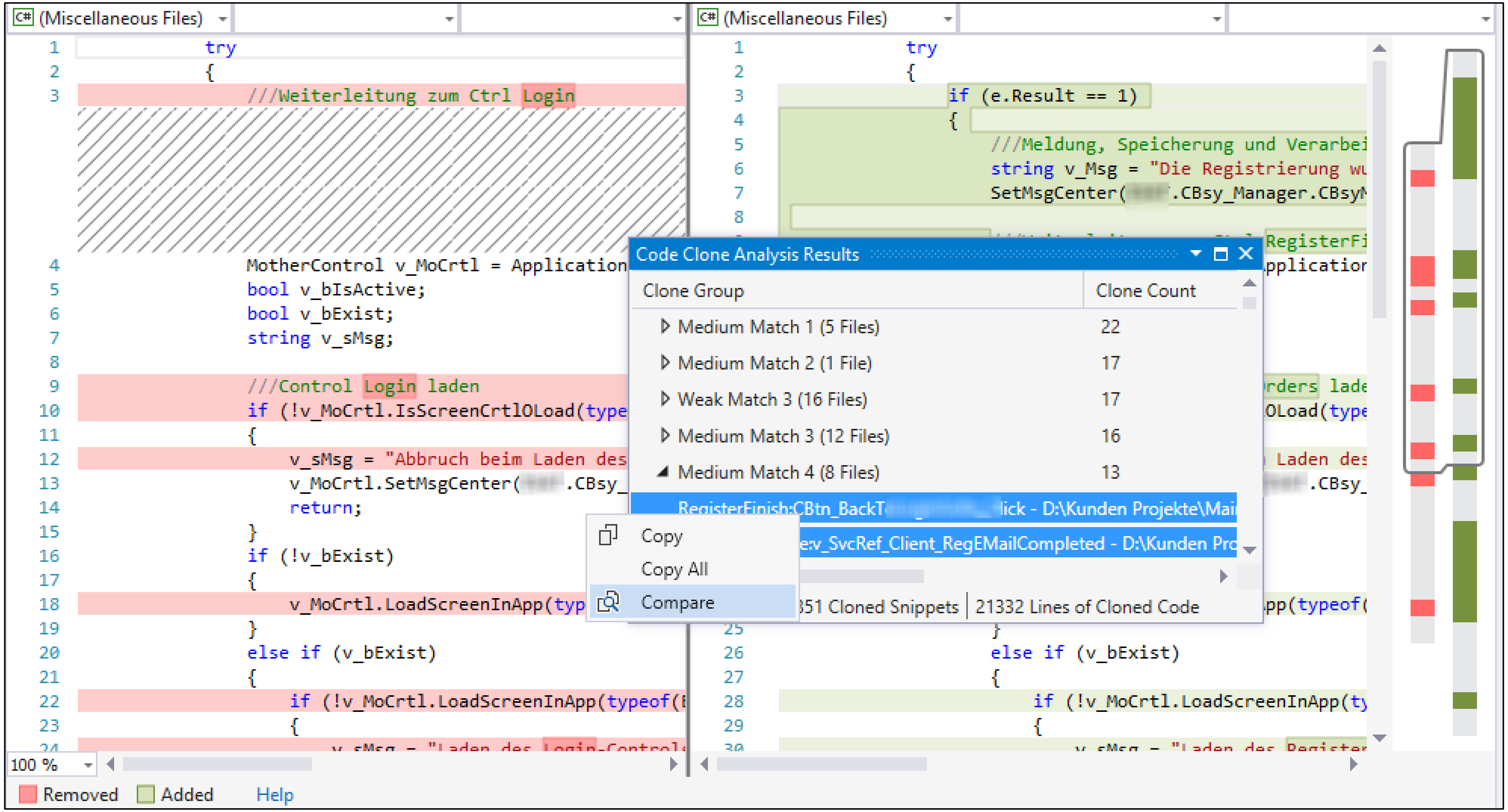
Code Clones vs. Duplikate



Code Clones vs. Duplikate



Code Clones vs. Duplikate



```
try
{
    ///Weiterleitung zum Ctrl Login

    MotherControl v_MoCrtl = Application
    bool v_bIsActive;
    bool v_bExist;
    string v_sMsg;

    ///Control Login laden
    if (!v_MoCrtl.IsScreenCrtlOLoad(type
    {
        v_sMsg = "Abbruch beim Laden des
        v_MoCrtl.SetMsgCenter([redacted].CBsy_
        return;
    }
    if (!v_bExist)
    {
        v_MoCrtl.LoadScreenInApp(typ
    }
    else if (v_bExist)
    {
        if (!v_MoCrtl.LoadScreenInApp(typeof(E
        {
            v_sMsg = "Laden des Login-Control
```

```
try
{
    if (e.Result == 1)
    {
        ///Meldung, Speicherung und Verarbei
        string v_Msg = "Die Registrierung wu
        SetMsgCenter([redacted].CBsy_Manager.CBsyM

                                   RegisterFi
                                   pplicatior

                                   rders lade
                                   OLoad(type

                                   Laden des
                          [redacted].CBsy_

                                   pp(typeof(
        }
        else if (v_bExist)
        {
            if (!v_MoCrtl.LoadScreenInApp(ty
            {
                v_sMsg = "Laden des Register
```

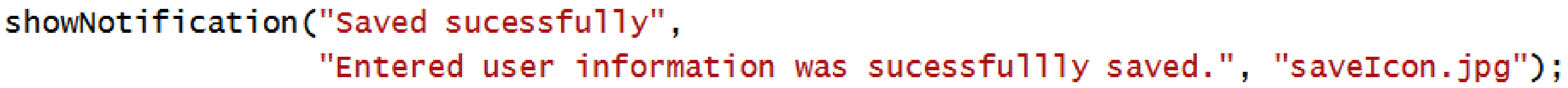
Code Clone Analysis Results

Clone Group	Clone Count
Medium Match 1 (5 Files)	22
Medium Match 2 (1 File)	17
Weak Match 3 (16 Files)	17
Medium Match 3 (12 Files)	16
Medium Match 4 (8 Files)	13
RegisterFinish:CBtn_BackT[redacted]lick - D:\Kunden Projekte\Mai	
e:v_SvcRef_Client_RegEMailCompleted - D:\Kunden Pro	

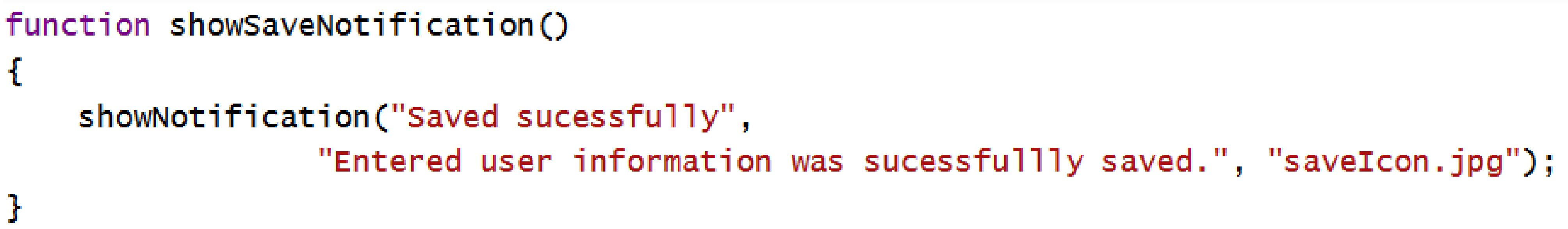
351 Cloned Snippets | 21332 Lines of Cloned Code

Code Clones vs. Duplikate

Don´t Repeat Yourself!!!



```
showNotification("Saved sucessfully",
                 "Entered user information was sucessfullly saved.", "saveIcon.jpg");
```



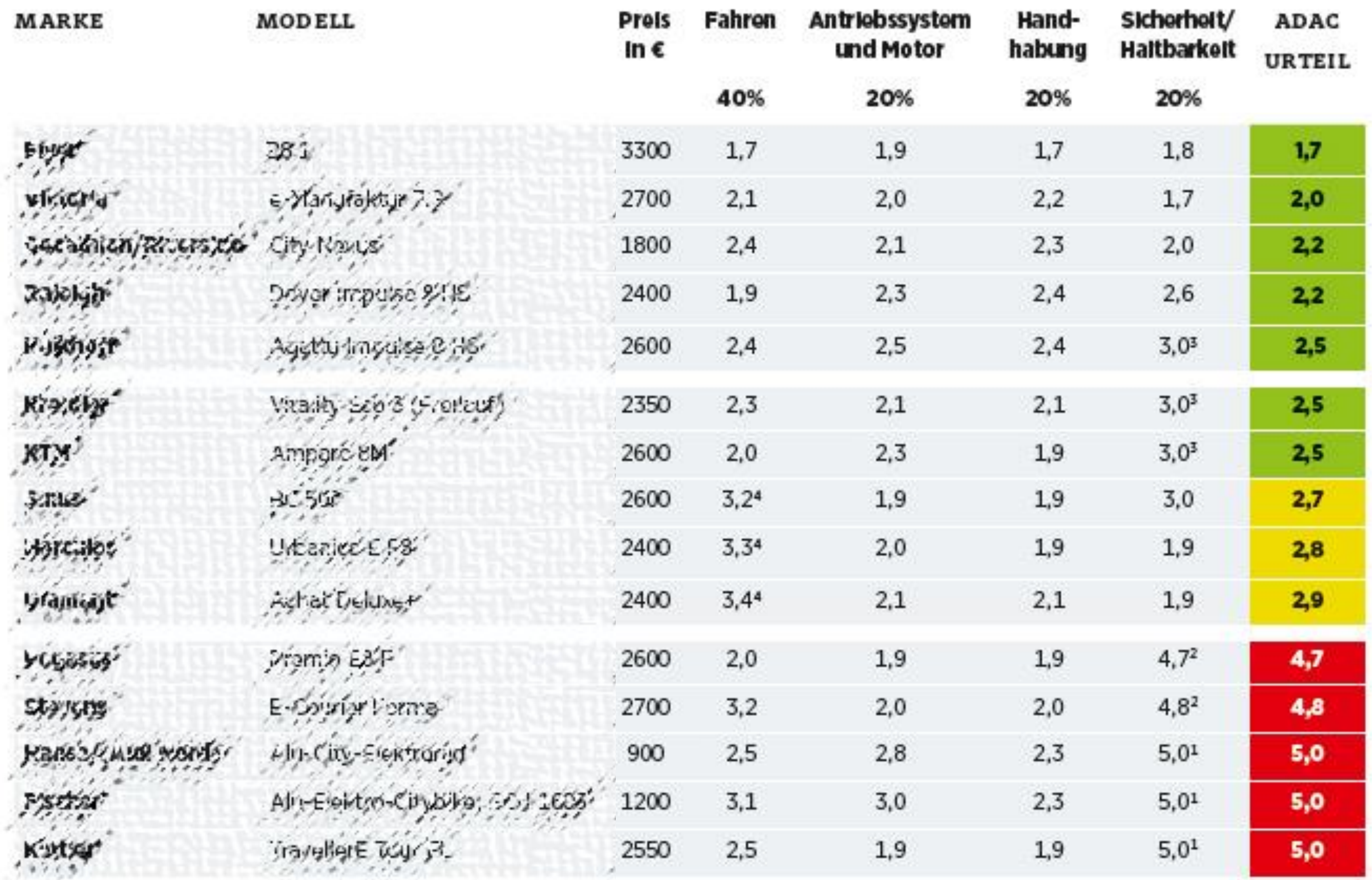
```
function showSaveNotification()
{
    showNotification("Saved sucessfully",
                 "Entered user information was sucessfullly saved.", "saveIcon.jpg");
}
```

Code Clones vs. Duplikate

```
public PageViewModel(INavigationService navigationService)
{
    if (navigationService == null)
    {
        throw new ArgumentException("navigationService");
    }

    this.NavigationService = navigationService;
}
```

Metriken

Metriken



MARKE	MODELL	Preis in €	Fahren	Antriebssystem und Motor	Hand-habung	Sicherheit/ Haltbarkeit	ADAC URTEIL
			40%	20%	20%	20%	
Flyer	28 i	3300	1,7	1,9	1,7	1,8	1,7
Victoria	e-Manufaktur 7.7	2700	2,1	2,0	2,2	1,7	2,0
Decathlon/Riverside	City Nexus	1800	2,4	2,1	2,3	2,0	2,2
Raleigh	Dover Impulse 9 HS	2400	1,9	2,3	2,4	2,6	2,2
Kalkhoff	Agattu Impulse 8 HS	2600	2,4	2,5	2,4	3,0[3]	2,5
Kreidler	Vitality Eco 8 (Freilauf)	2350	2,3	2,1	2,1	3,0[3]	2,5
KTM	Amparo 8M	2600	2,0	2,3	1,9	3,0[3]	2,5
Sinus	BC 50F	2600	3,2[4]	1,9	1,9	3,0	2,7
Hercules	Urbanico E F8	2400	3,3[4]	2,0	1,9	1,9	2,8
Diamant	Achat Deluxe+	2400	3,4[4]	2,1	2,1	1,9	2,9
Pegasus	Premio E8F	2600	2,0	1,9	1,9	4,7[2]	4,7
Stevens	E-Courier Forma	2700	3,2	2,0	2,0	4,8[2]	4,8
Rex/Aldi Nord	Alu-City-Elektrorad	900	2,5	2,8	2,3	5,0[1]	5,0
Fischer	Alu-Elektro-Citybike, ECU 1603	1200	3,1	3,0	2,3	5,0[1]	5,0
Kettler	Traveller E Tour FL	2550	2,5	1,9	1,9	5,0[1]	5,0

Lines of Code

	SLOC (in Mio.)
Windows NT 3.1 (1993)	4–5
Windows NT 3.5 (1994)	7–8
Windows NT 4.0 (1996)	11–12
Windows 2000 (2000)	> 29
Windows XP (2001)	40
Windows Server 2003 (2003)	50

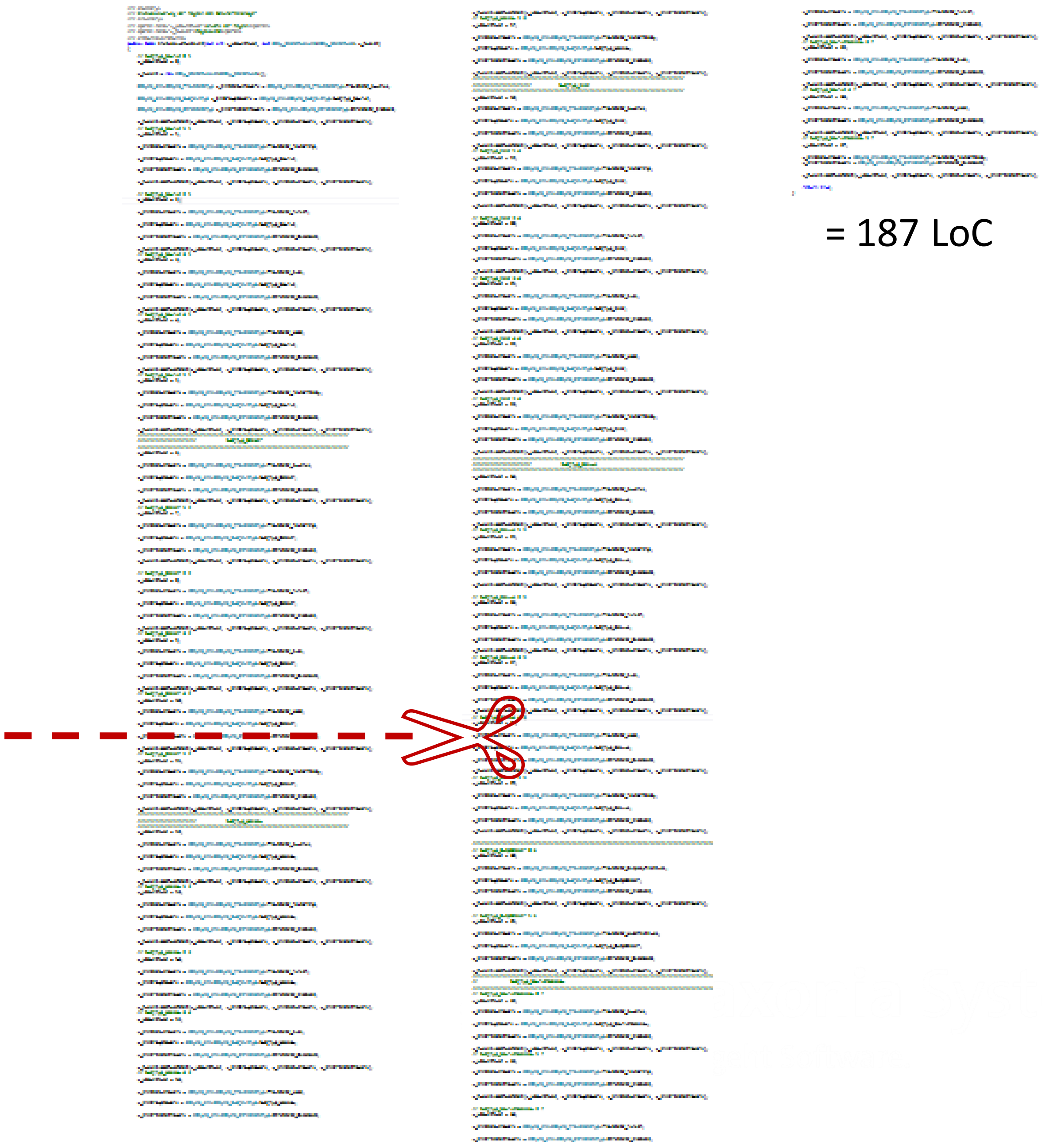
Lines of Code

Lines of Code

π • 👍

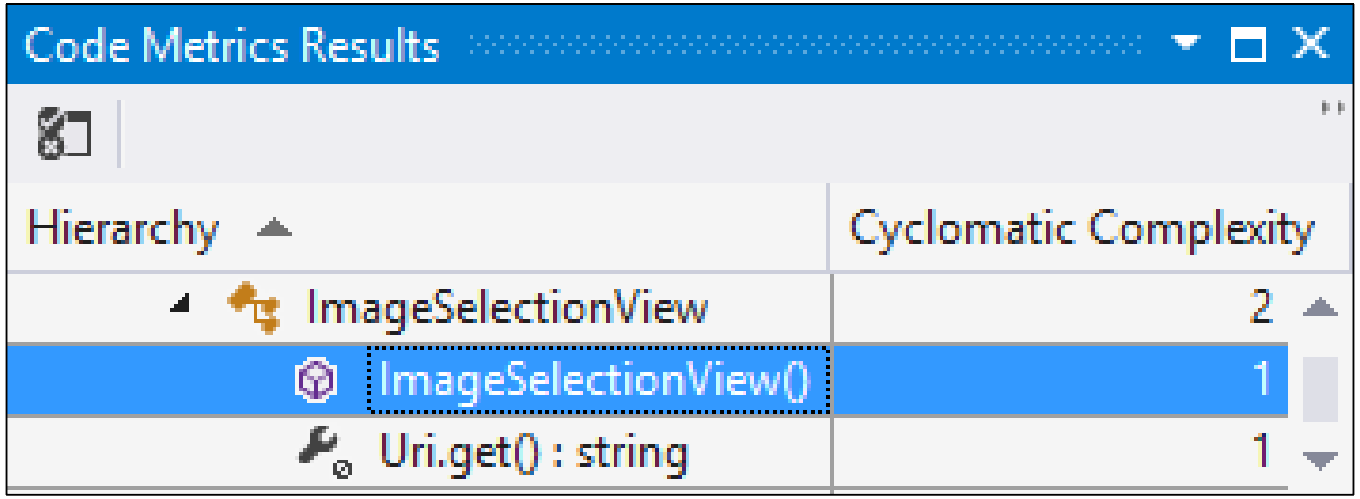
Klasse < 500 LoC
Methode < 50 LoC

?

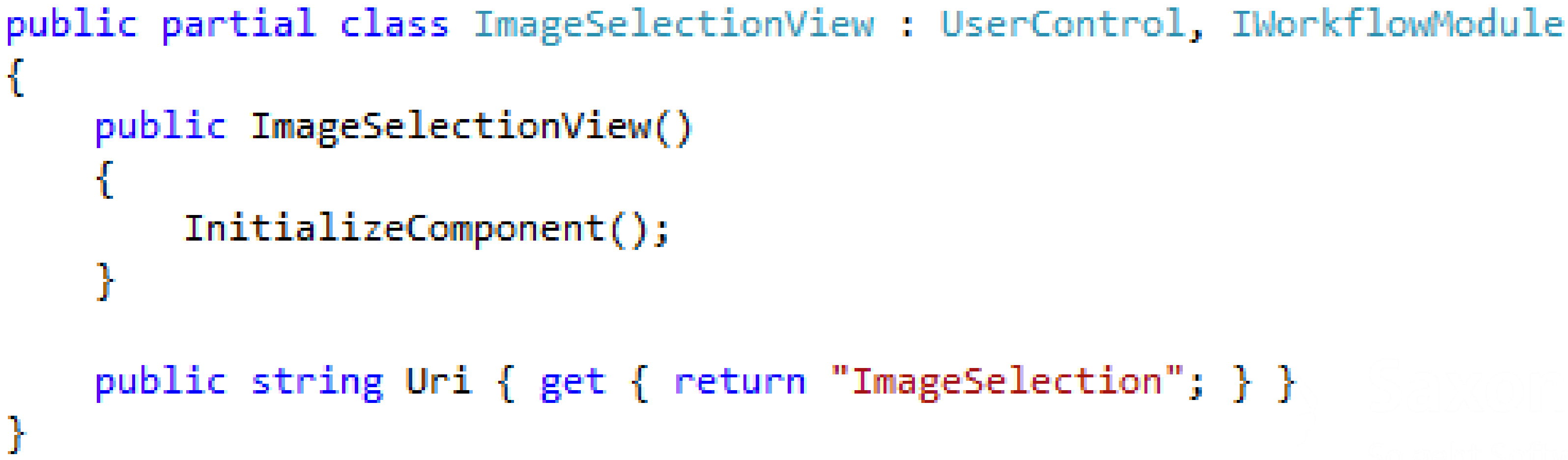


= 187 LoC

Cyclomatic Complexity

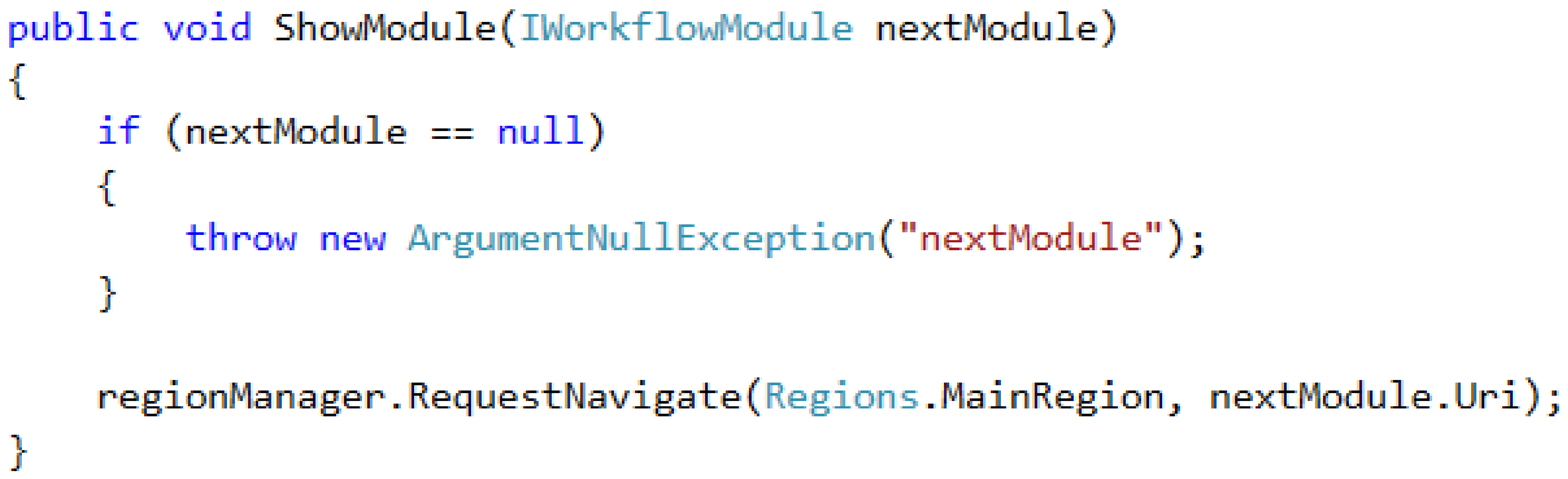


Hierarchy	Cyclomatic Complexity
ImageSelectionView	2
ImageSelectionView()	1
Uri.get() : string	1



```
public partial class ImageSelectionView : UserControl, IWorkflowModule
{
    public ImageSelectionView()
    {
        InitializeComponent();
    }

    public string Uri { get { return "ImageSelection"; } }
}
```

Cyclomatic Complexity



```
public void ShowModule(IWorkflowModule nextModule)
{
    if (nextModule == null)
    {
        throw new ArgumentNullException("nextModule");
    }

    regionManager.RequestNavigate(Regions.MainRegion, nextModule.Uri);
}
```

Komplexität == 2

Cyclomatic Complexity

π • 👍

für Typen

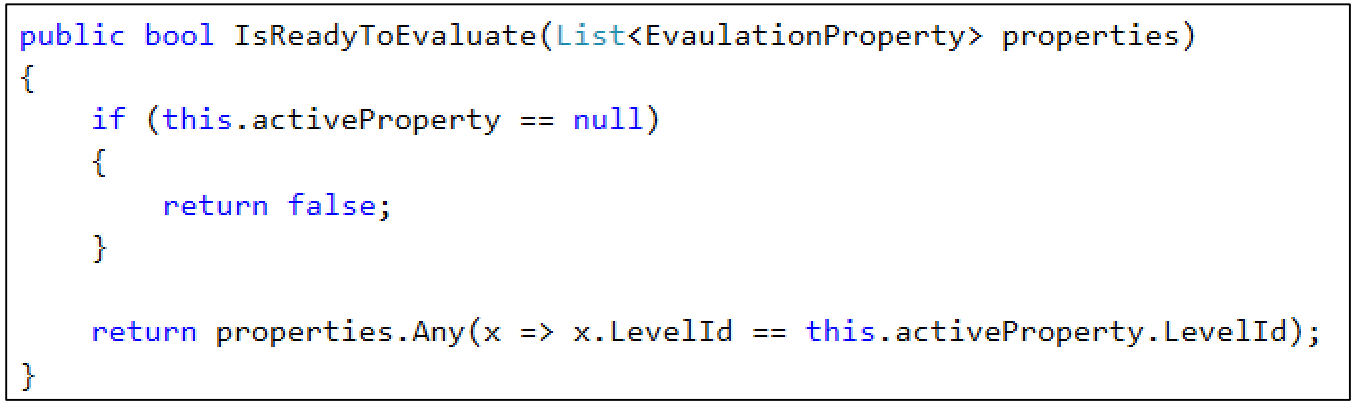
1 bis 10 = normal

11 bis 20 = moderat

21 bis 50 = riskant

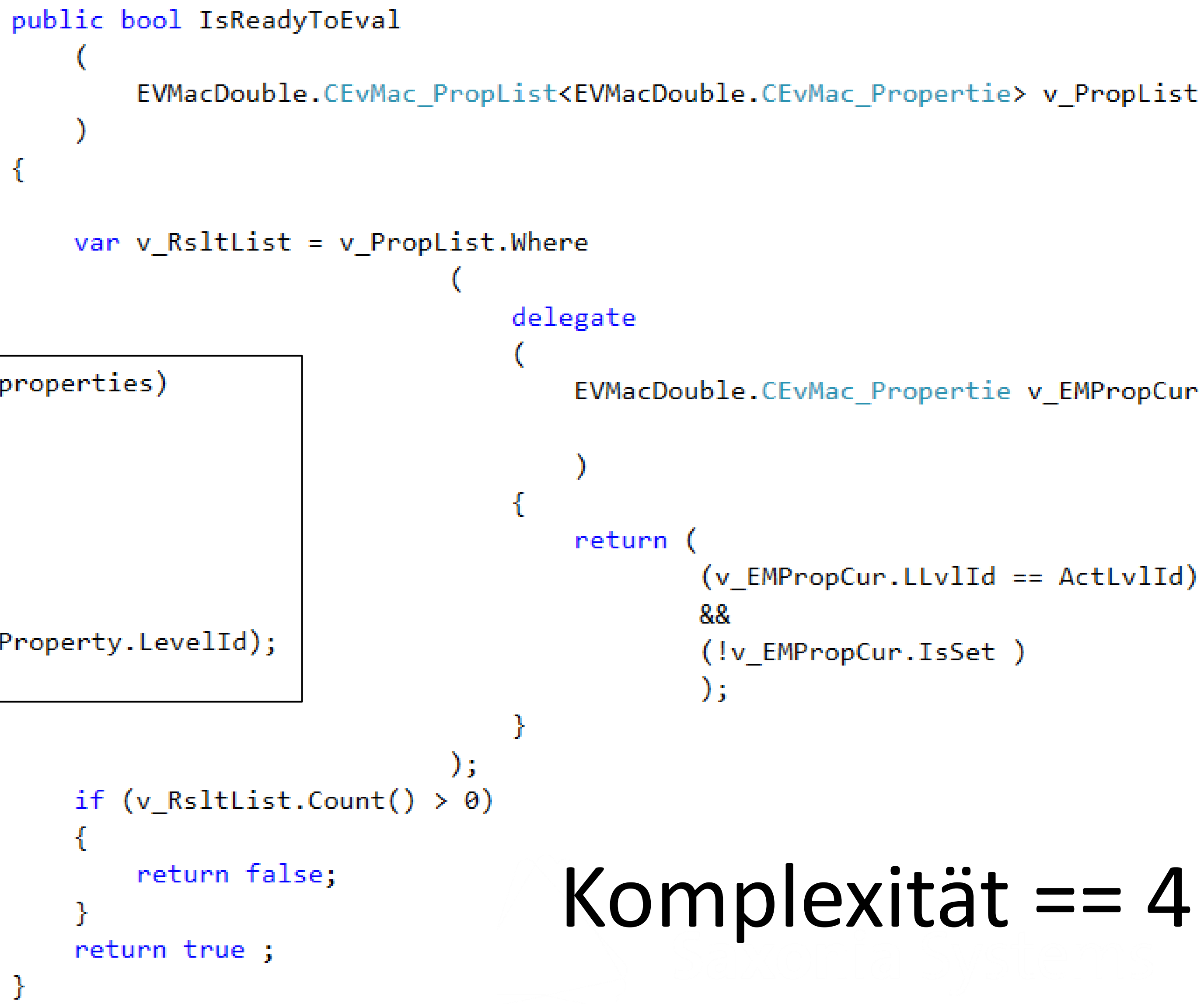
> 50 = instabil

Cyclomatic Complexity



```
public bool IsReadyToEvaluate(List<EvaulationProperty> properties)
{
    if (this.activeProperty == null)
    {
        return false;
    }

    return properties.Any(x => x.LevelId == this.activeProperty.LevelId);
}
```

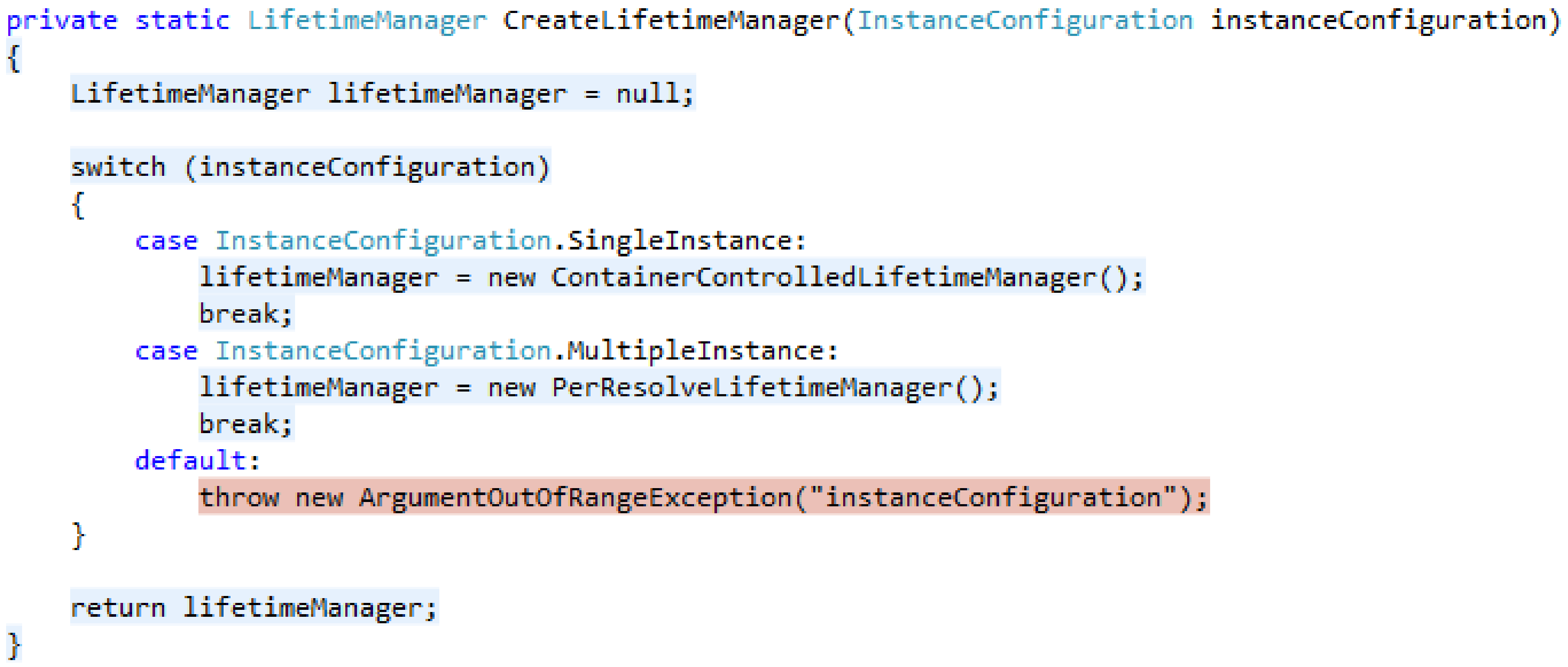
Komplexität == 4



```
public bool IsReadyToEval
    (
        EVMacDouble.CEvMac_PropList<EVMacDouble.CEvMac_Propertie> v_PropList
    )
{

    var v_RsltList = v_PropList.Where
                            (
                                delegate
                                (
                                    EVMacDouble.CEvMac_Propertie v_EMPropCur

                                    )
                                {
                                    return (
                                            (v_EMPropCur.LLvlId == ActLvlId)
                                            &&
                                            (!v_EMPropCur.IsSet )
                                            );
                                }
                            );
    if (v_RsltList.Count() > 0)
    {
        return false;
    }
    return true ;
}
```

Komplexität == 4

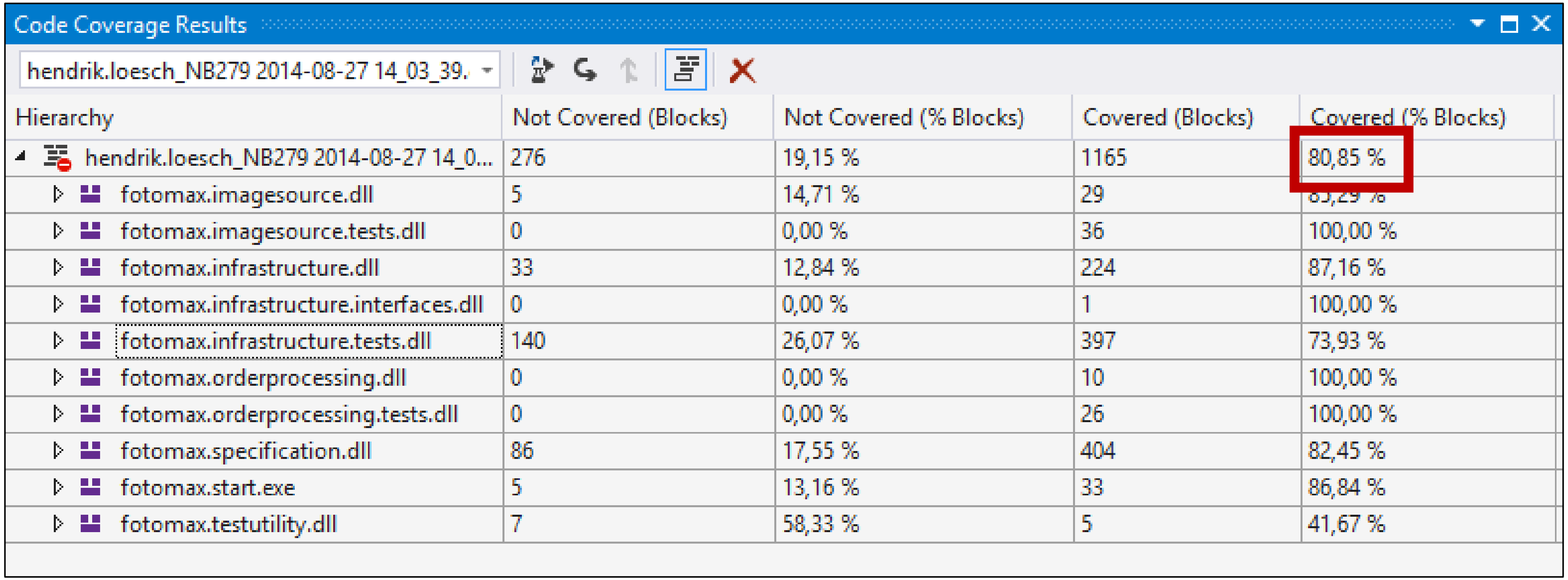
Code Coverage aka Test Coverage



```
private static LifetimeManager CreateLifetimeManager(InstanceConfiguration instanceConfiguration)
{
    LifetimeManager lifetimeManager = null;

    switch (instanceConfiguration)
    {
        case InstanceConfiguration.SingleInstance:
            lifetimeManager = new ContainerControlledLifetimeManager();
            break;
        case InstanceConfiguration.MultipleInstance:
            lifetimeManager = new PerResolveLifetimeManager();
            break;
        default:
            throw new ArgumentOutOfRangeException("instanceConfiguration");
    }

    return lifetimeManager;
}
```

Komplexität == 3

Code Coverage aka Test Coverage



Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks)
hendrik.loesch_NB279 2014-08-27 14_0...	276	19,15 %	1165	80,85 %
fotomax.imagesource.dll	5	14,71 %	29	85,29 %
fotomax.imagesource.tests.dll	0	0,00 %	36	100,00 %
fotomax.infrastructure.dll	33	12,84 %	224	87,16 %
fotomax.infrastructure.interfaces.dll	0	0,00 %	1	100,00 %
fotomax.infrastructure.tests.dll	140	26,07 %	397	73,93 %
fotomax.orderprocessing.dll	0	0,00 %	10	100,00 %
fotomax.orderprocessing.tests.dll	0	0,00 %	26	100,00 %
fotomax.specification.dll	86	17,55 %	404	82,45 %
fotomax.start.exe	5	13,16 %	33	86,84 %
fotomax.testutility.dll	7	58,33 %	5	41,67 %

C.R.A.P.

C.R.A.P.(m) = CC(m)^2 * (1 – Coverage(m)/100)^3 + CC(m)

> “The C.R.A.P. (Change Risk Analysis and Predictions) index is designed to analyze and predict the amount of effort, pain, and time required to maintain an existing body of code.”
>
> Alberto Savoia

Quelle: http://www.artima.com/weblogs/viewpost.jsp?thread=210575

C.R.A.P.

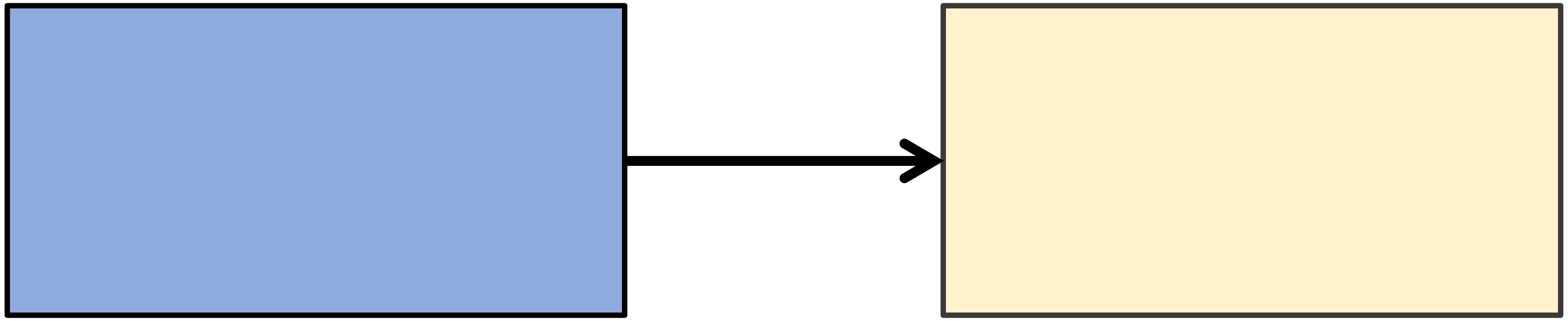
C.R.A.P.(m) = CC(m)^2 * (1 – Coverage(m)/100)^3 + CC(m)

π • 👍
30

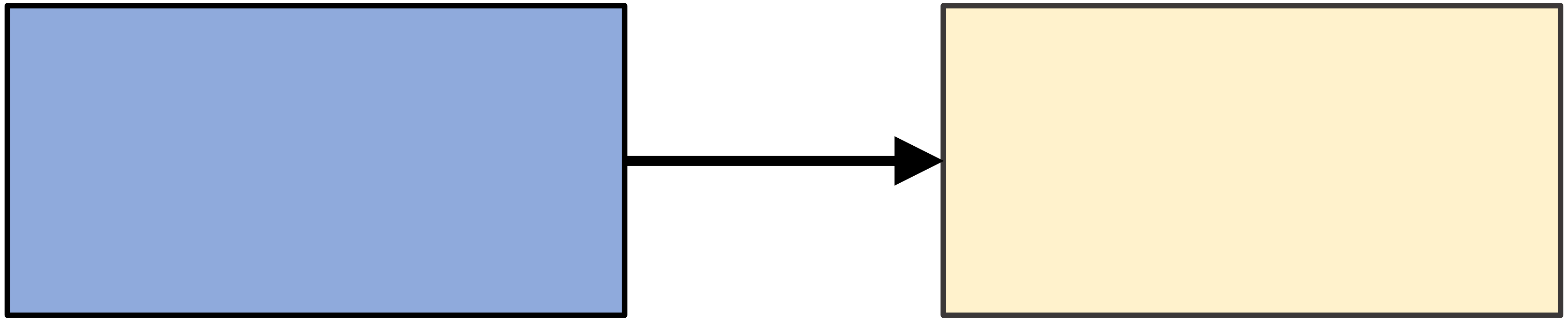
Method’s CC	% of coverage required to be below CRAPpy threshold
0 – 5	0%
10	42%
15	57%
20	71%
25	80%
30	100%
31+	No amount of testing will keep methods this complex out of CRAP territory.

Quelle: https://www.artima.com/weblogs/viewpost.jsp?thread=215899

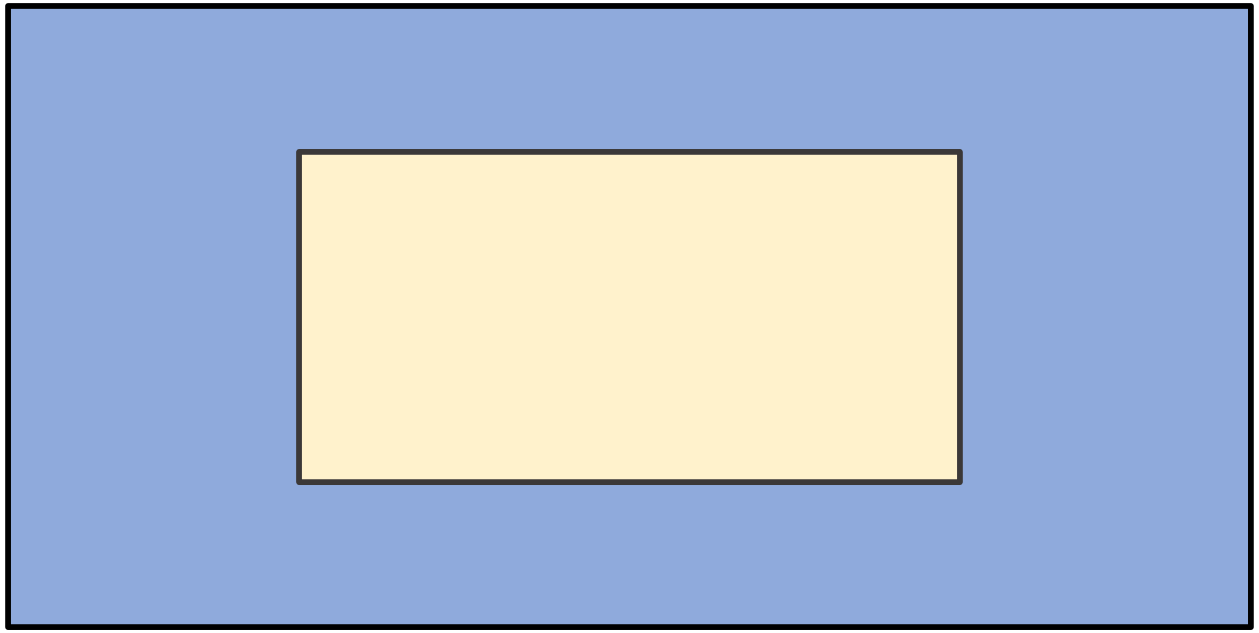
Depth of Inheritance



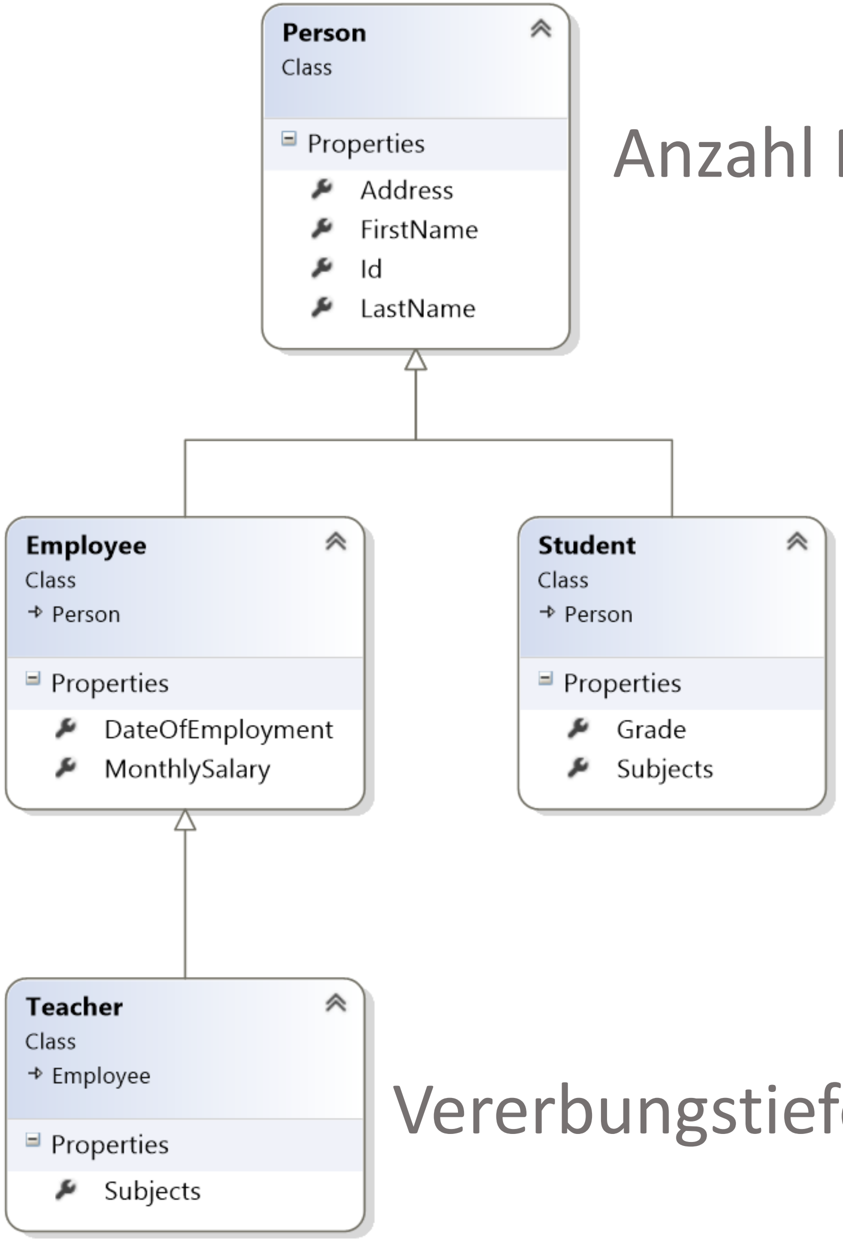
Depth of Inheritance



Depth of Inheritance



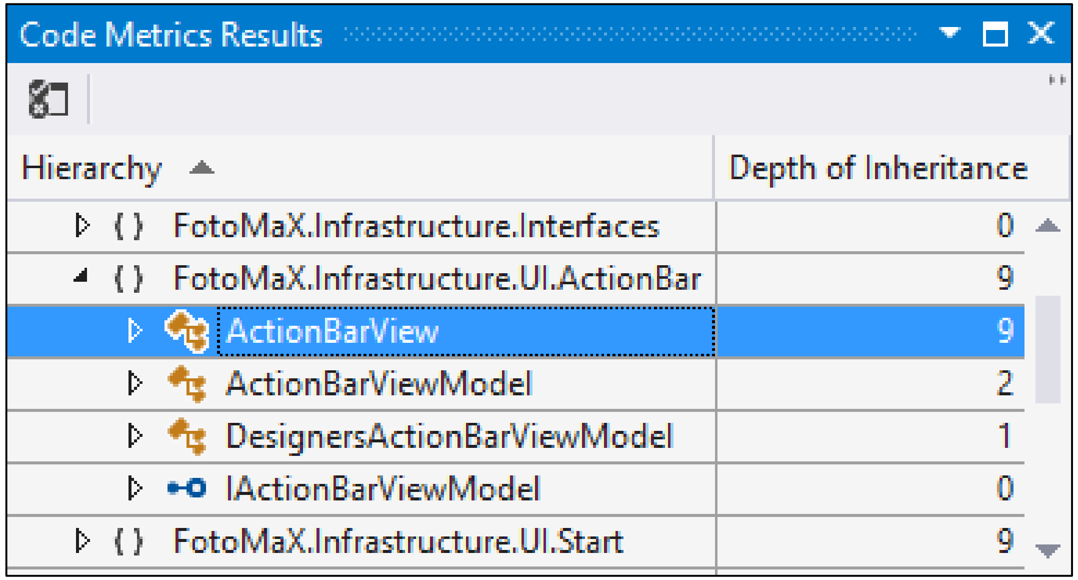
Depth of Inheritance



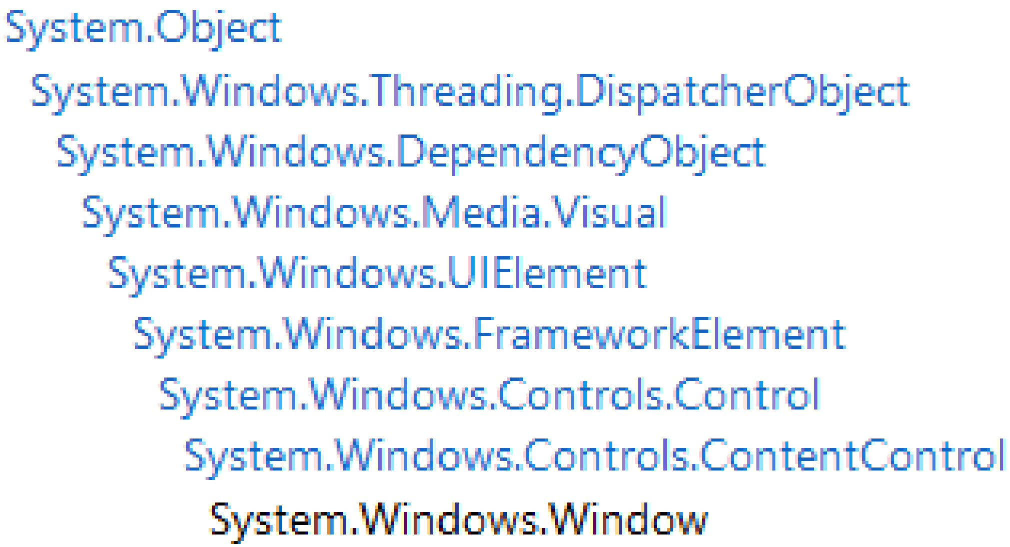
Anzahl Nachfolger = 3

Vererbungstiefe = 2

Depth of Inheritance



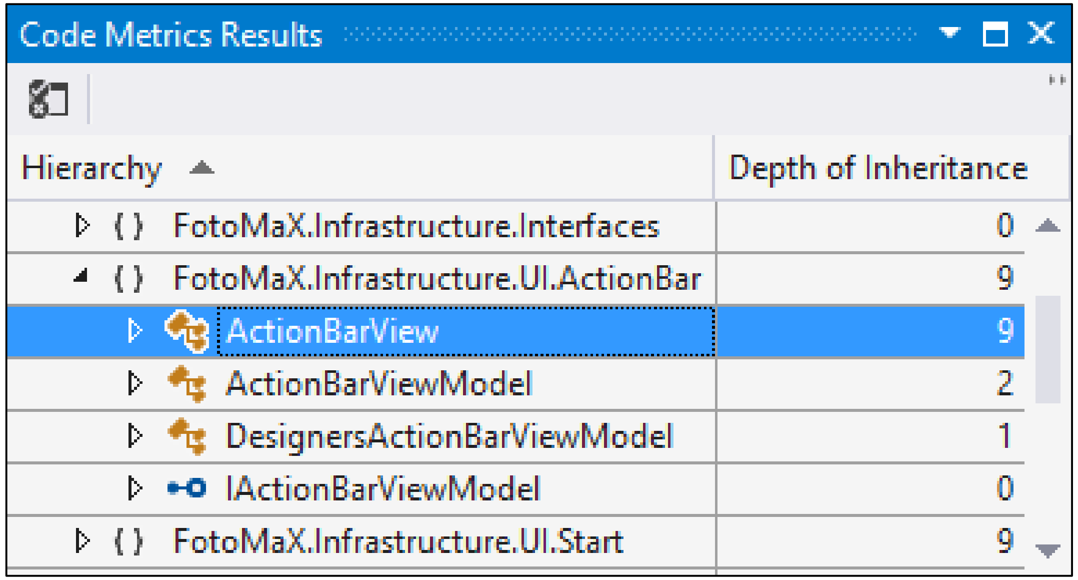
Hierarchy	Depth of Inheritance
{ } FotoMaX.Infrastructure.Interfaces	0
{ } FotoMaX.Infrastructure.UI.ActionBar	9
ActionBarView	9
ActionBarViewModel	2
DesignersActionBarViewModel	1
IActionBarViewModel	0
{ } FotoMaX.Infrastructure.UI.Start	9



```
System.Object
  System.Windows.Threading.DispatcherObject
    System.Windows.DependencyObject
      System.Windows.Media.Visual
        System.Windows.UIElement
          System.Windows.FrameworkElement
            System.Windows.Controls.Control
              System.Windows.Controls.ContentControl
                System.Windows.Window
```

Depth of Inheritance

Favour Composition over Inheritance!!!



Code Metrics Results

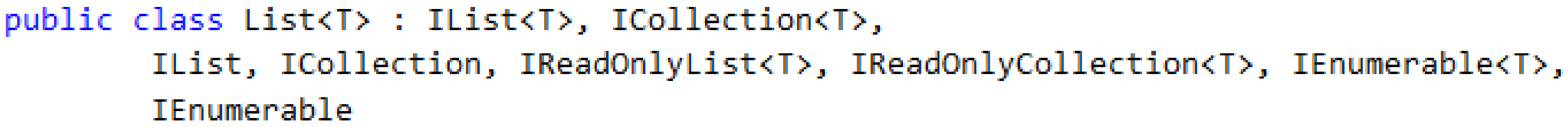
Hierarchy	Depth of Inheritance
FotoMaX.Infrastructure.Interfaces	0
FotoMaX.Infrastructure.UI.ActionBar	9
ActionBarView	9
ActionBarViewModel	2
DesignersActionBarViewModel	1
IActionBarViewModel	0
FotoMaX.Infrastructure.UI.Start	9

π • 👍

Hierachieebene <= 6

```
public class ActionBarViewModel : ReactiveObject, IActionBarViewModel
```

Depth of Inheritance



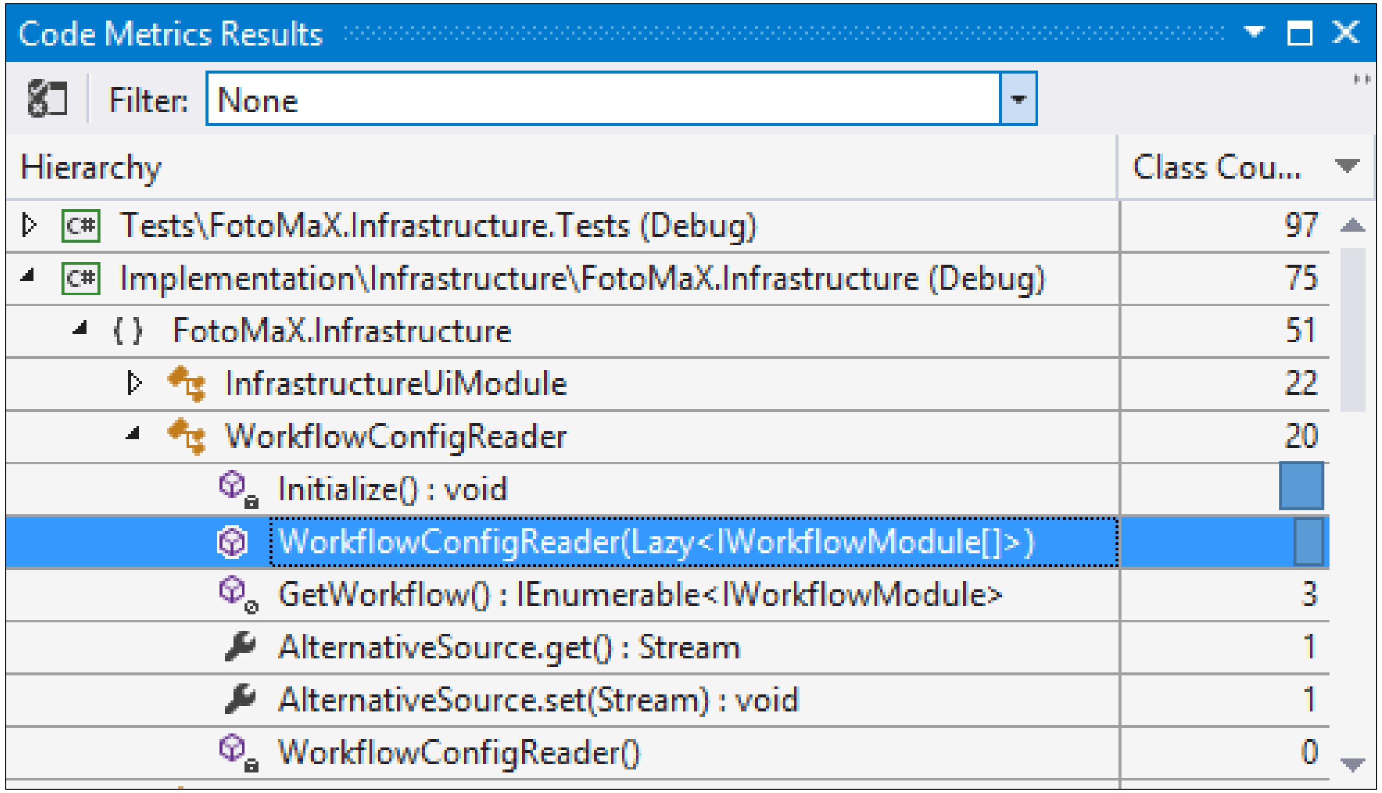
```
public class List<T> : IList<T>, ICollection<T>,
        IList, ICollection, IReadOnlyList<T>, IReadOnlyCollection<T>, IEnumerable<T>,
        IEnumerable
```

- ✓ Information Hiding Principle
- ✓ Interface Segregation Principle

Class Coupling

1 2 3

```
public WorkflowConfigReader(Lazy<IWorkflowModule[]> availableWorkflowModules)
{
    this.availableWorkflowModules = availableWorkflowModules;
}
```

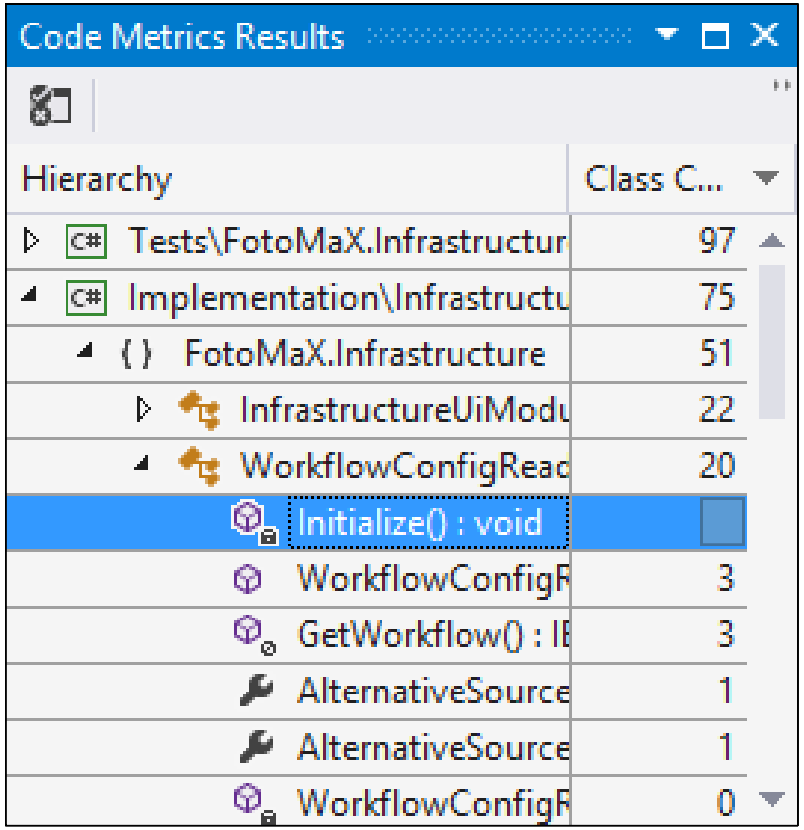


Code Metrics Results

Filter: None

Hierarchy	Class Cou...
Tests\FotoMaX.Infrastructure.Tests (Debug)	97
Implementation\Infrastructure\FotoMaX.Infrastructure (Debug)	75
{ } FotoMaX.Infrastructure	51
InfrastructureUiModule	22
WorkflowConfigReader	20
Initialize() : void	
WorkflowConfigReader(Lazy<IWorkflowModule[]>)	
GetWorkflow() : IEnumerable<IWorkflowModule>	3
AlternativeSource.get() : Stream	1
AlternativeSource.set(Stream) : void	1
WorkflowConfigReader()	0

Class Coupling



Hierarchy	Class C...
Tests\FotoMaX.Infrastructur	97
Implementation\Infrastructu	75
{ } FotoMaX.Infrastructure	51
InfrastructureUiModu	22
WorkflowConfigRead	20
Initialize() : void	
WorkflowConfigR	3
GetWorkflow() : IE	3
AlternativeSource	1
AlternativeSource	1
WorkflowConfigR	0

```
private void Initialize()
{
    configuredWorkflow.Clear();

    XmlReader configReader;
    if (this.AlternativeSource != null)
    {
        configReader = XmlReader.Create(this.AlternativeSource);
    }
    else
    {
        var path = Environment.CurrentDirectory + defaultPath;
        if (!File.Exists(path))
        {
            throw new InvalidOperationException("Workflow configuration is missing.");
        }

        configReader = XmlReader.Create(File.OpenRead(path));
    }

    var root = XElement.Load(configReader);
    foreach (var configuredModules in root.Descendants("workstep"))
    {
        var uri = configuredModules.Attribute("uri").Value;

        var availableModule = this.availableWorkflowModules.Value.FirstOrDefault(x => x.Uri == uri);
        if (availableModule != null)
        {
            this.configuredWorkflow.Add(availableModule);
        }
    }
}
```

Single Responsibility Principle???

Fazit

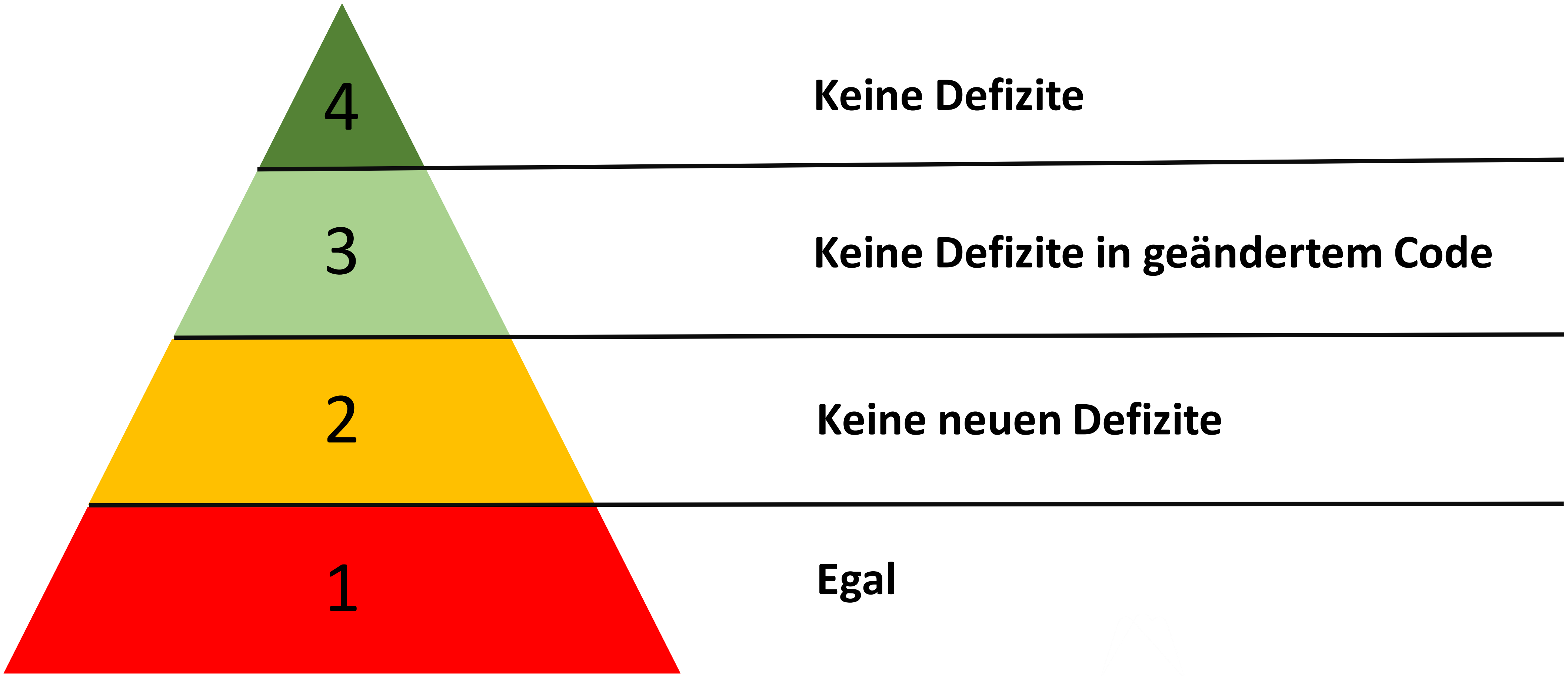
Premature Optimization

„The real problem is that programmers have spent far too much time worrying about efficiency in the wrong places and at the wrong times; premature optimization is the root of all evil (or at least most of it) in programming.“

Donald Knuth

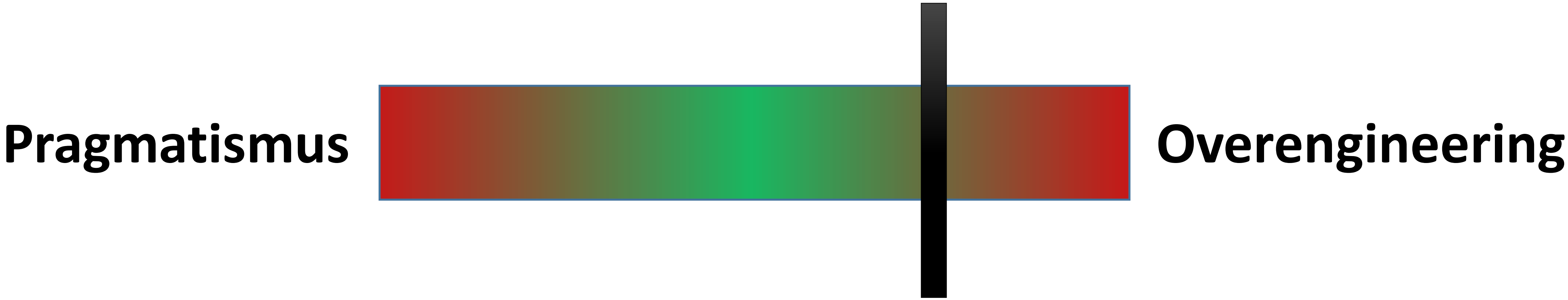
Quelle: https://en.wikiquote.org/wiki/Donald_Knuth

Wieviel ist gut für mich?



Quelle: Dr. Elmar Jürgens, CQSE

Pragmatismus vs. Overengineering



Pragmatismus

Overengineering

Der Sprecher

Hendrik Lösch

Senior Consultant

Coach

@HerrLoesch

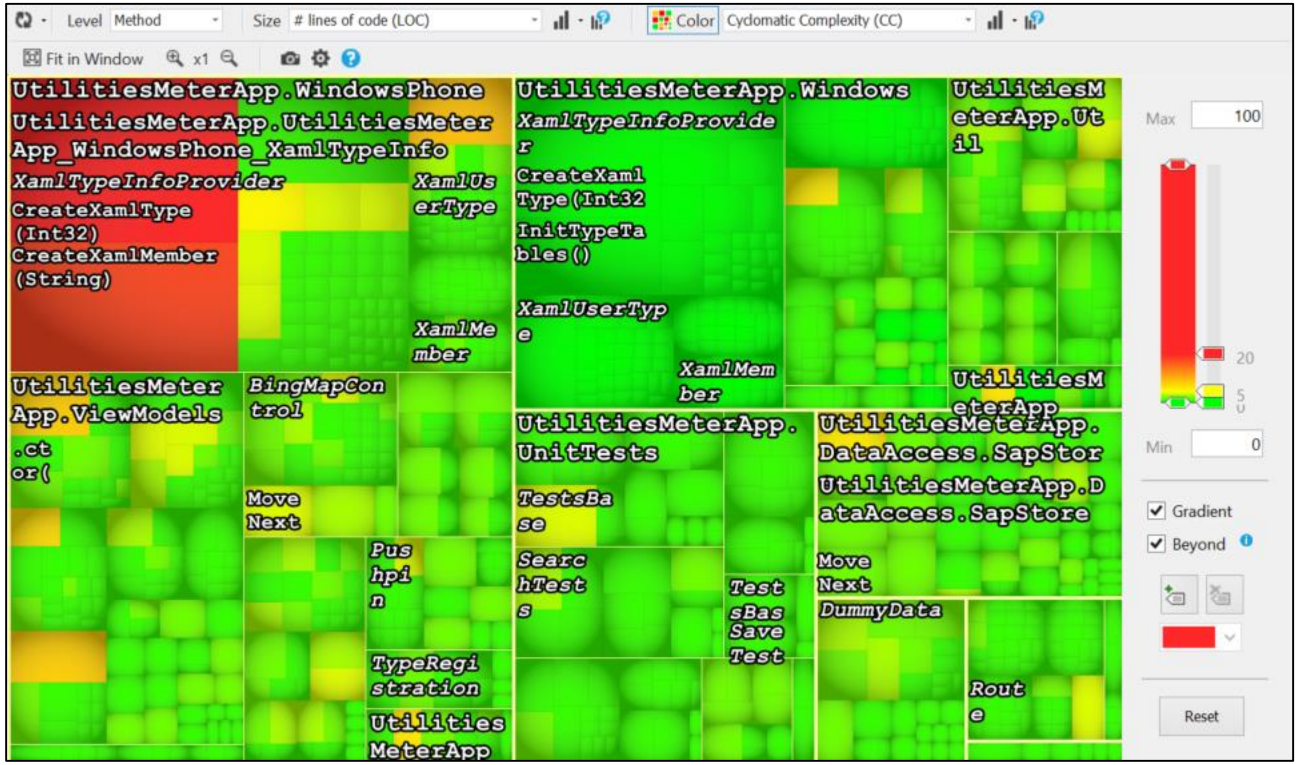
Hendrik.Loesch@saxsys.de

Just-About.Net

Saxonia Systems
So geht Software.

Tools

NDepend



Level: Method

Size: # lines of code (LOC)

Color: Cyclomatic Complexity (CC)

Max: 100

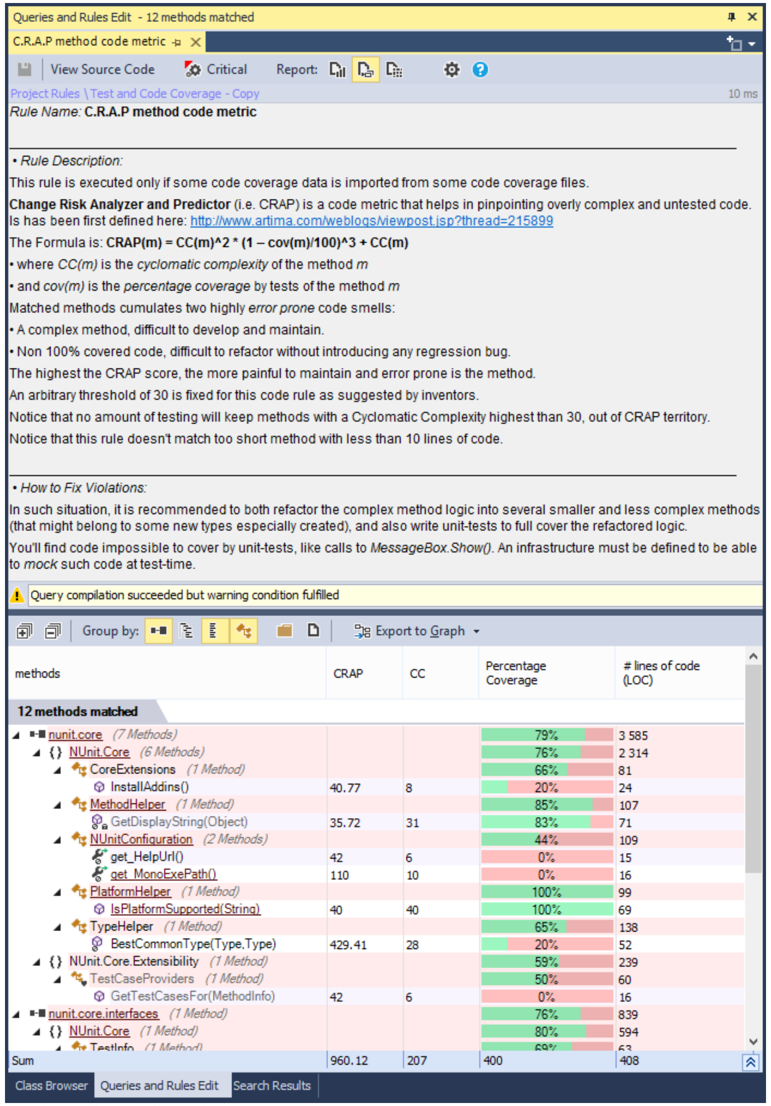
Min: 0

☑ Gradient

☑ Beyond

Heatmap

http://www.ndepend.com/



Queries and Rules Edit - 12 methods matched

C.R.A.P method code metric

Project Rules \ Test and Code Coverage - Copy

10 ms

Rule Name: **C.R.A.P method code metric**

- *Rule Description:*

This rule is executed only if some code coverage data is imported from some code coverage files.

Change Risk Analyzer and Predictor (i.e. CRAP) is a code metric that helps in pinpointing overly complex and untested code. Is has been first defined here: http://www.artima.com/weblogs/viewpost.jsp?thread=215899

The Formula is: **CRAP(m) = CC(m)^2 * (1 – cov(m)/100)^3 + CC(m)**

- where *CC(m)* is the *cyclomatic complexity* of the method *m*
- and *cov(m)* is the *percentage coverage* by tests of the method *m*

Matched methods cumulates two highly *error prone* code smells:

- A complex method, difficult to develop and maintain.
- Non 100% covered code, difficult to refactor without introducing any regression bug.

The highest the CRAP score, the more painful to maintain and error prone is the method.

An arbitrary threshold of 30 is fixed for this code rule as suggested by inventors.

Notice that no amount of testing will keep methods with a Cyclomatic Complexity highest than 30, out of CRAP territory.

Notice that this rule doesn't match too short method with less than 10 lines of code.

- *How to Fix Violations:*

In such situation, it is recommended to both refactor the complex method logic into several smaller and less complex methods (that might belong to some new types especially created), and also write unit-tests to full cover the refactored logic.

You'll find code impossible to cover by unit-tests, like calls to *MessageBox.Show()*. An infrastructure must be defined to be able to *mock* such code at test-time.

⚠ Query compilation succeeded but warning condition fulfilled

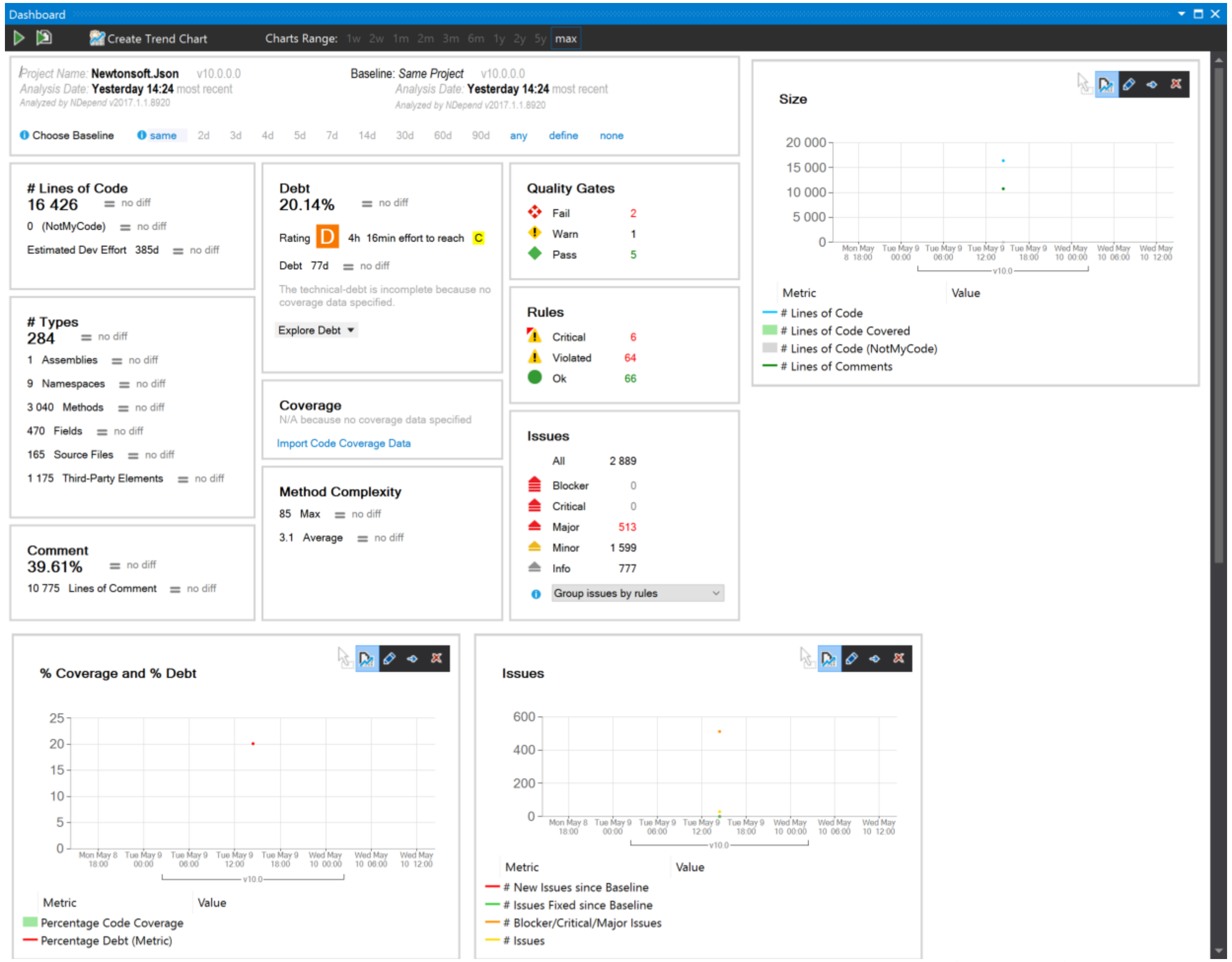
Group by: Export to Graph

12 methods matched

methods	CRAP	CC	Percentage Coverage	# lines of code (LOC)
nunit.core (7 Methods)			79%	3 585
NUnit.Core (6 Methods)			76%	2 314
CoreExtensions (1 Method)			66%	81
InstallAddins()	40.77	8	20%	24
MethodHelper (1 Method)			85%	107
GetDisplayString(Object)	35.72	31	83%	71
NUnitConfiguration (2 Methods)			44%	109
get_HelpUrl()	42	6	0%	15
get_MonoExePath()	110	10	0%	16
PlatformHelper (1 Method)			100%	99
IsPlatformSupported(String)	40	40	100%	69
TypeHelper (1 Method)			65%	138
BestCommonType(Type,Type)	429.41	28	20%	52
NUnit.Core.Extensibility (1 Method)			59%	239
TestCaseProviders (1 Method)			50%	60
GetTestCasesFor(MethodInfo)	42	6	0%	16
nunit.core.interfaces (1 Method)			76%	839
NUnit.Core (1 Method)			80%	594
TestInfo (1 Method)			69%	63
Sum	960.12	207	400	408

Qualitätsregeln inkl. Erklärungen

NDepend



Project Name: **Newtonsoft.Json** v10.0.0.0
Analysis Date: **Yesterday 14:24** most recent
Analyzed by NDepend v2017.1.1.8920

Baseline: *Same Project* v10.0.0.0
Analysis Date: **Yesterday 14:24** most recent
Analyzed by NDepend v2017.1.1.8920

Choose Baseline same 2d 3d 4d 5d 7d 14d 30d 60d 90d any define none

# Lines of Code
16 426 = no diff
0 (NotMyCode) = no diff
Estimated Dev Effort 385d = no diff

# Types
284 = no diff
1 Assemblies = no diff
9 Namespaces = no diff
3 040 Methods = no diff
470 Fields = no diff
165 Source Files = no diff
1 175 Third-Party Elements = no diff

Comment
39.61% = no diff
10 775 Lines of Comment = no diff

Debt
20.14% = no diff
Rating D 4h 16min effort to reach C
Debt 77d = no diff
The technical-debt is incomplete because no coverage data specified.
Explore Debt ▾

Coverage
N/A because no coverage data specified
Import Code Coverage Data

Method Complexity
85 Max = no diff
3.1 Average = no diff

Quality Gates

Fail	2
Warn	1
Pass	5

Rules

Critical	6
Violated	64
Ok	66

Issues

All	2 889
Blocker	0
Critical	0
Major	513
Minor	1 599
Info	777

Group issues by rules

Einschätzung der technischen Schuld und Darstellung des Trends

xDepend

VBDepend

CppDepend

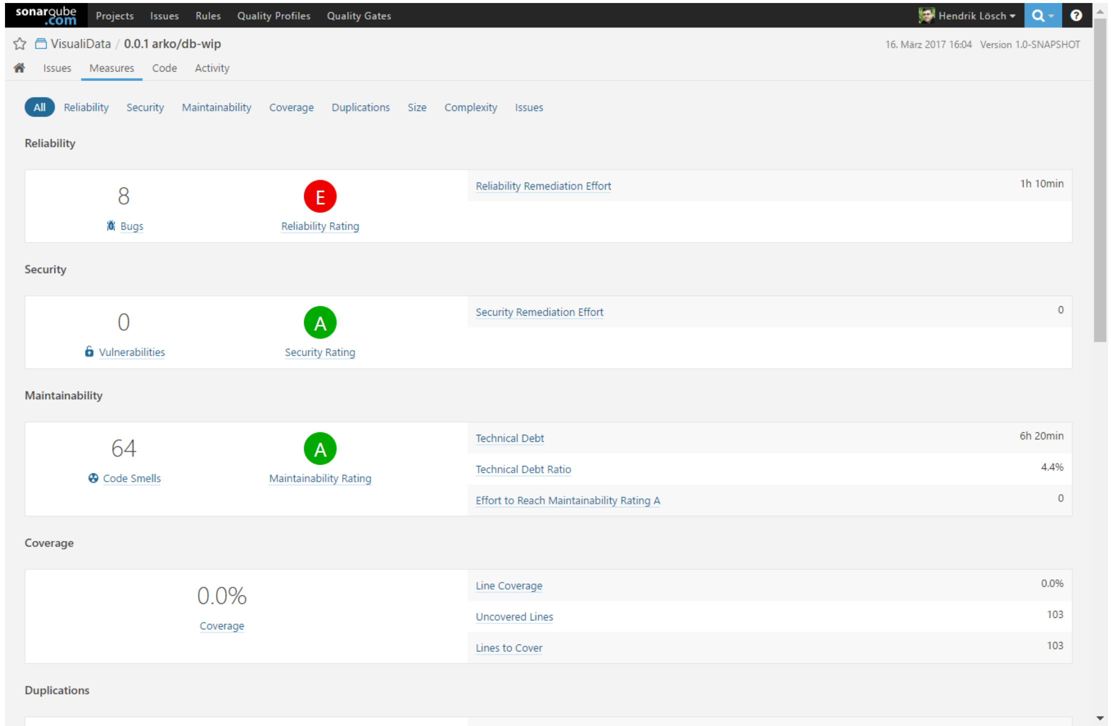
Auch verfügbar für andere Sprachen und Technologiestacks

JArchitect

XClarify

SonarQube

- Analyse der Codebasis auf Basis von Standardregeln
- Buildintegration für Java, JavaScript, C# C/C++, …



VisualiData / 0.0.1 arko/db-wip

16. März 2017 16:04 Version 1.0-SNAPSHOT

Reliability

8 Bugs — E Reliability Rating

Reliability Remediation Effort	1h 10min

Security

0 Vulnerabilities — A Security Rating

Security Remediation Effort	0

Maintainability

64 Code Smells — A Maintainability Rating

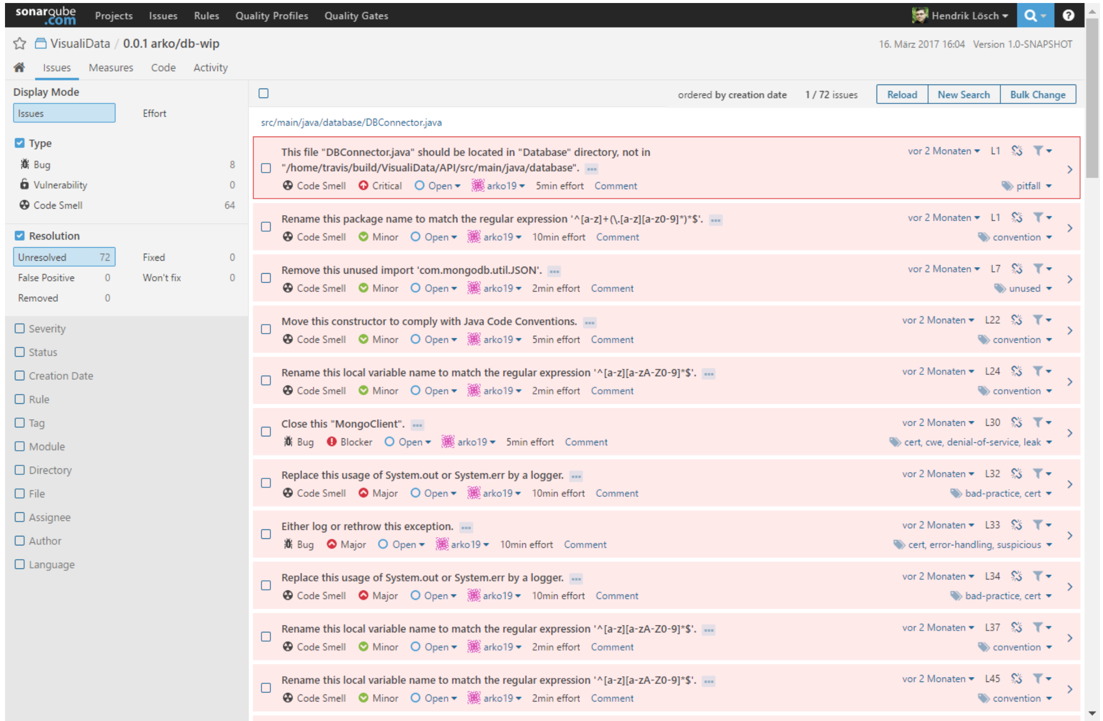
Technical Debt	6h 20min
Technical Debt Ratio	4.4%
Effort to Reach Maintainability Rating A	0

Coverage

0.0% Coverage

Line Coverage	0.0%
Uncovered Lines	103
Lines to Cover	103

Duplications



VisualiData / 0.0.1 arko/db-wip

16. März 2017 16:04 Version 1.0-SNAPSHOT

Display Mode: Issues | Effort

☑ Type

Bug	8
Vulnerability	0
Code Smell	64

☑ Resolution

Unresolved	72	Fixed	0
False Positive	0	Won't fix	0
Removed	0		

☐ Severity
☐ Status
☐ Creation Date
☐ Rule
☐ Tag
☐ Module
☐ Directory
☐ File
☐ Assignee
☐ Author
☐ Language

ordered by creation date 1 / 72 issues

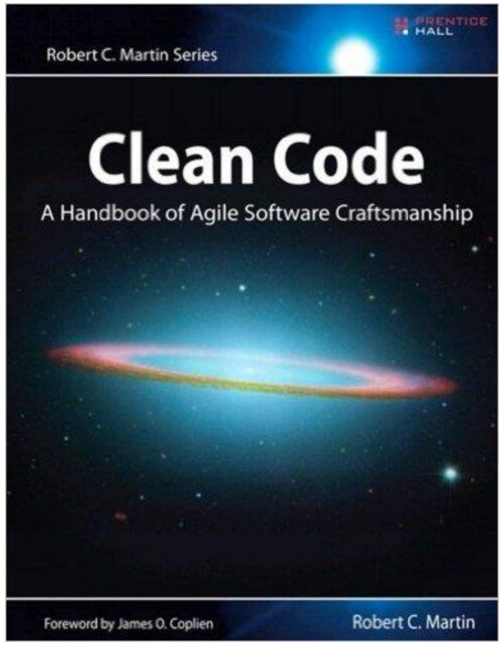
src/main/java/database/DBConnector.java

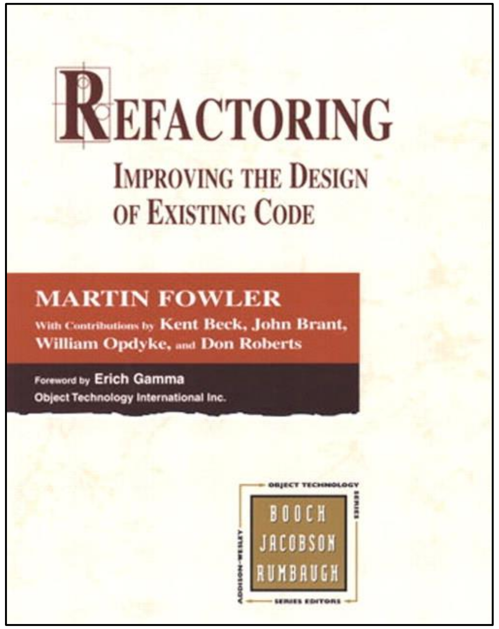
- This file "DBConnector.java" should be located in "Database" directory, not in "/home/travis/build/VisualiData/API/src/main/java/database". — Code Smell, Critical, Open, arko19, 5min effort, Comment — vor 2 Monaten, L1 — pitfall
- Rename this package name to match the regular expression '^[a-z]+(\.[a-z][a-z0-9]*)*$'. — Code Smell, Minor, Open, arko19, 10min effort, Comment — vor 2 Monaten, L1 — convention
- Remove this unused import 'com.mongodb.util.JSON'. — Code Smell, Minor, Open, arko19, 2min effort, Comment — vor 2 Monaten, L7 — unused
- Move this constructor to comply with Java Code Conventions. — Code Smell, Minor, Open, arko19, 5min effort, Comment — vor 2 Monaten, L22 — convention
- Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'. — Code Smell, Minor, Open, arko19, 2min effort, Comment — vor 2 Monaten, L24 — convention
- Close this "MongoClient". — Bug, Blocker, Open, arko19, 5min effort, Comment — vor 2 Monaten, L30 — cert, cwe, denial-of-service, leak
- Replace this usage of System.out or System.err by a logger. — Code Smell, Major, Open, arko19, 10min effort, Comment — vor 2 Monaten, L32 — bad-practice, cert
- Either log or rethrow this exception. — Bug, Major, Open, arko19, 10min effort, Comment — vor 2 Monaten, L33 — cert, error-handling, suspicious
- Replace this usage of System.out or System.err by a logger. — Code Smell, Major, Open, arko19, 10min effort, Comment — vor 2 Monaten, L34 — bad-practice, cert
- Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'. — Code Smell, Minor, Open, arko19, 2min effort, Comment — vor 2 Monaten, L37 — convention
- Rename this local variable name to match the regular expression '^[a-z][a-zA-Z0-9]*$'. — Code Smell, Minor, Open, arko19, 2min effort, Comment — vor 2 Monaten, L45 — convention

https://www.sonarqube.org/

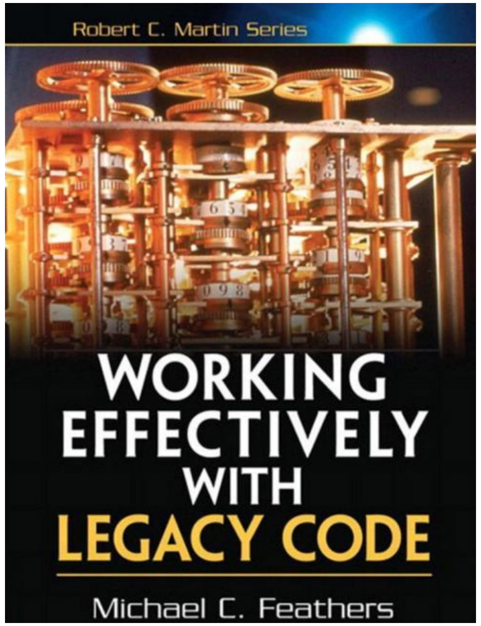
Literatur

Literatur


PRENTICE HALL
Robert C. Martin Series
Clean Code
A Handbook of Agile Software Craftsmanship
Foreword by James O. Coplien
Robert C. Martin


REFACTORING
IMPROVING THE DESIGN OF EXISTING CODE
MARTIN FOWLER
With Contributions by Kent Beck, John Brant, William Opdyke, and Don Roberts
Foreword by Erich Gamma
Object Technology International Inc.
OBJECT TECHNOLOGY SERIES
BOOCH JACOBSON RUMBAUGH
ADDISON-WESLEY
SERIES EDITORS

refactoring.com



Carola Lilienthal
Langlebige Software-Architekturen
Wie man Komplexität in den Griff bekommt
dpunkt.verlag