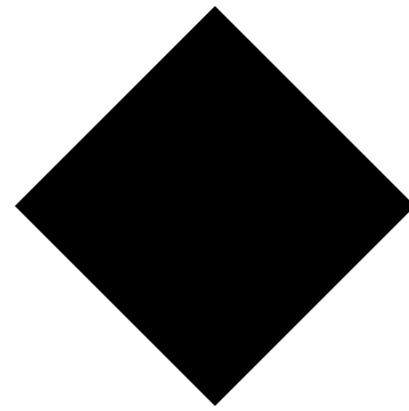


Beyond the Concept of Quality in *Pharo*

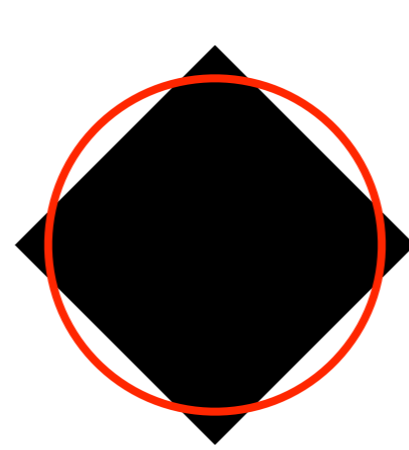
Yuriy Tymchuk | Michele Lanza



Circle

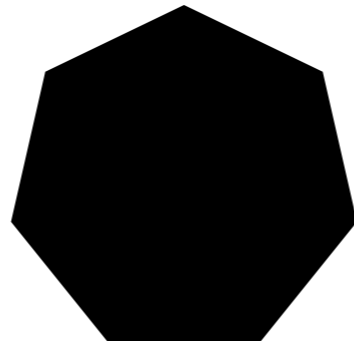


Circle

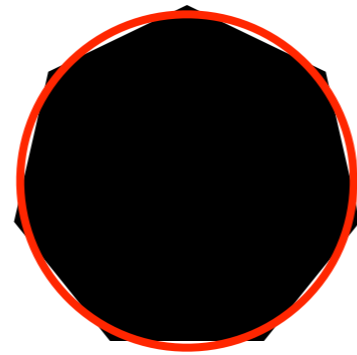


$$x^2 + y^2 = r^2$$

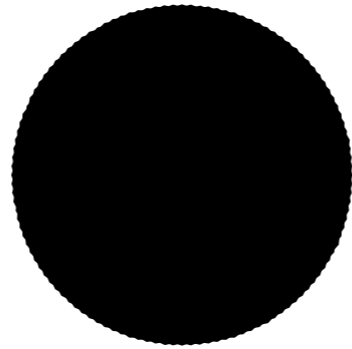
Circle



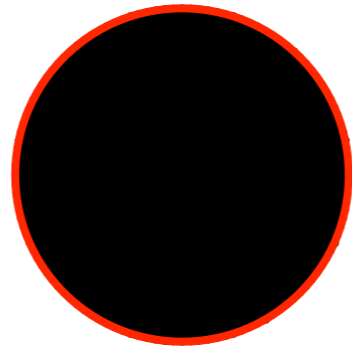
Circle



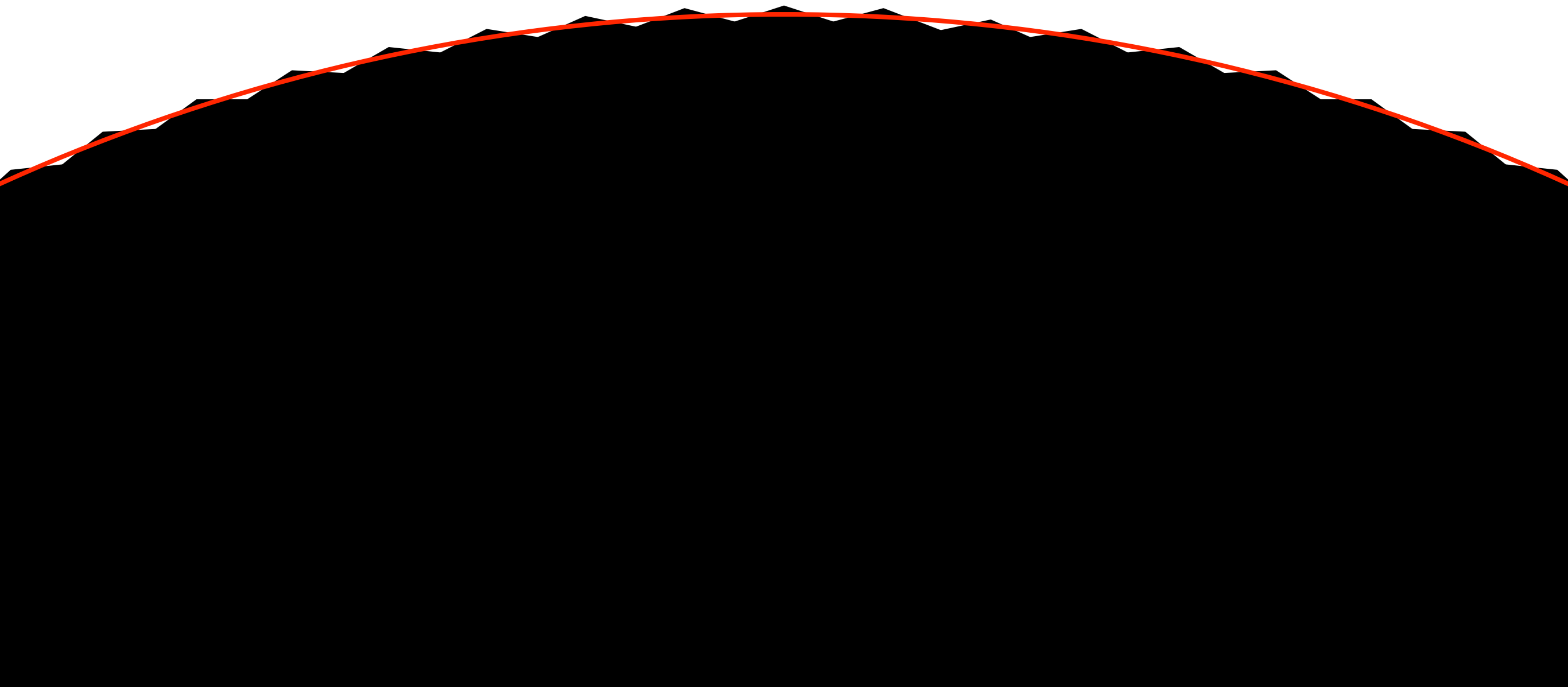
Circle



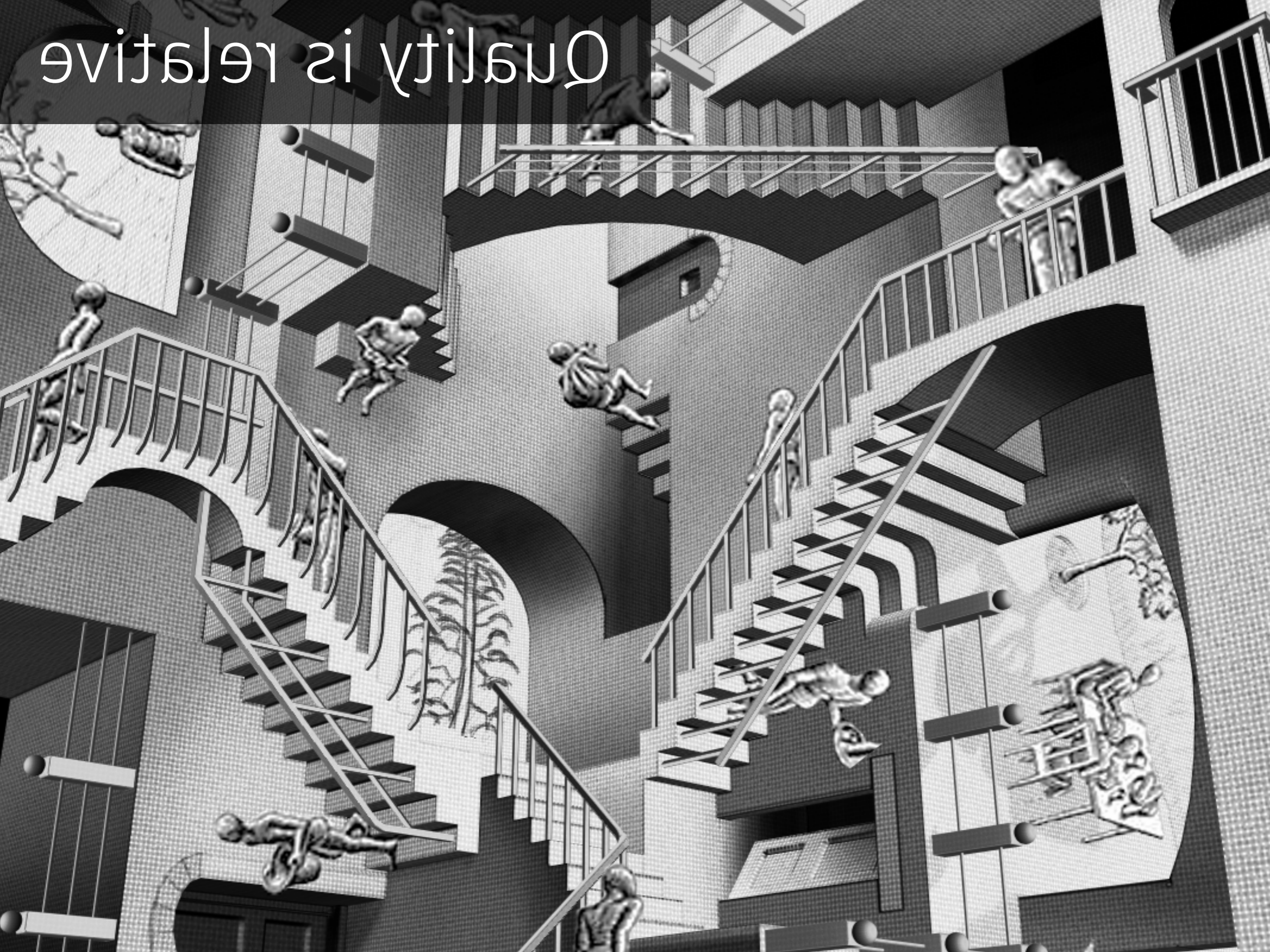
Circle



Circle

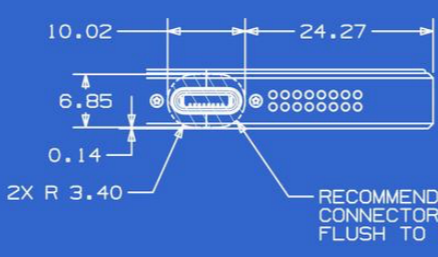
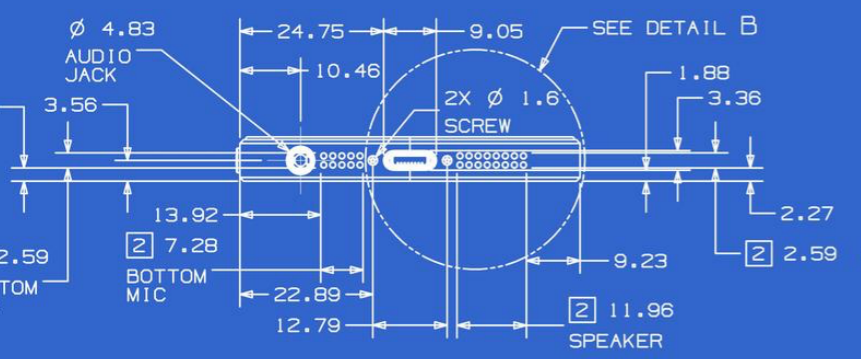
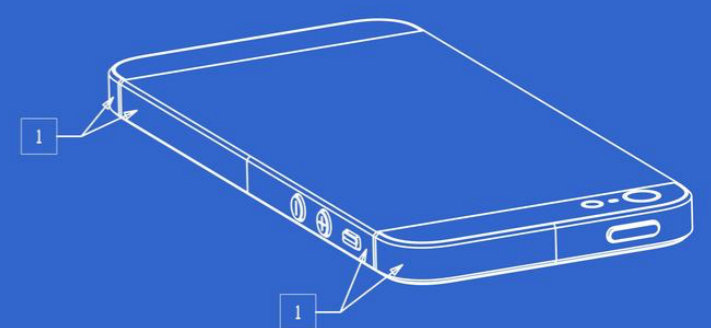
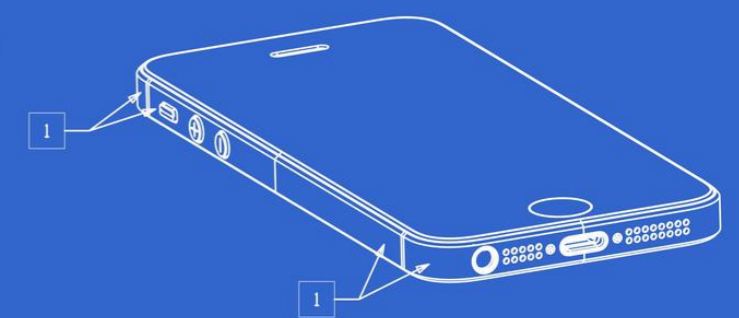
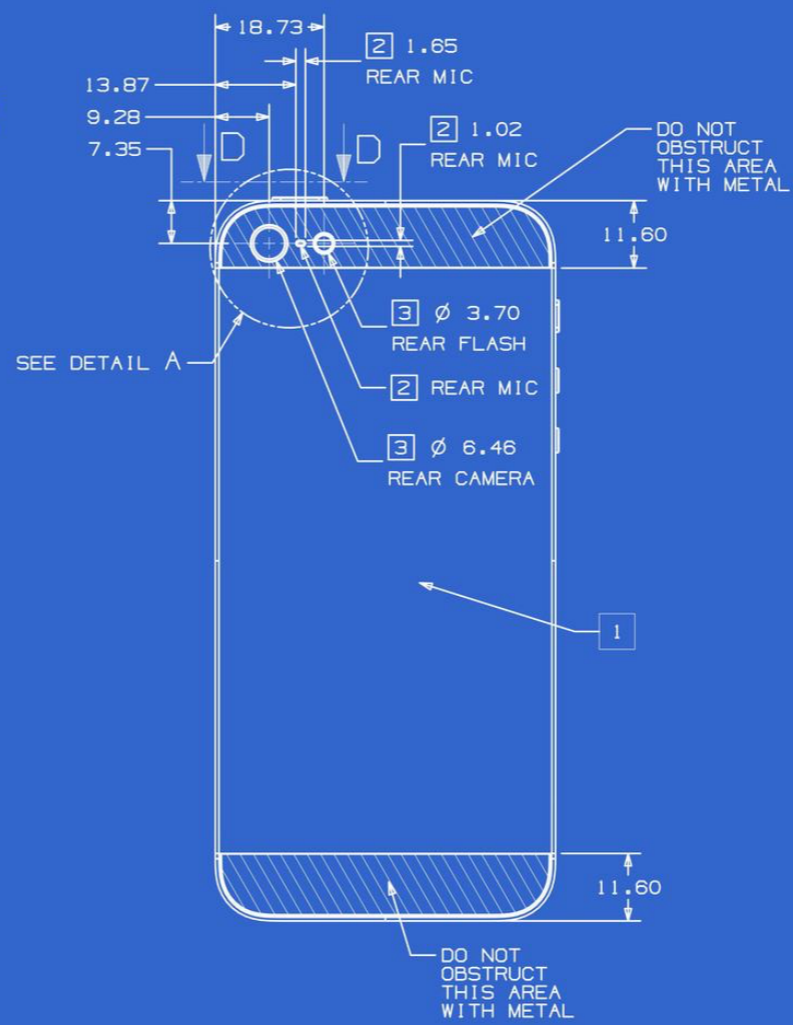
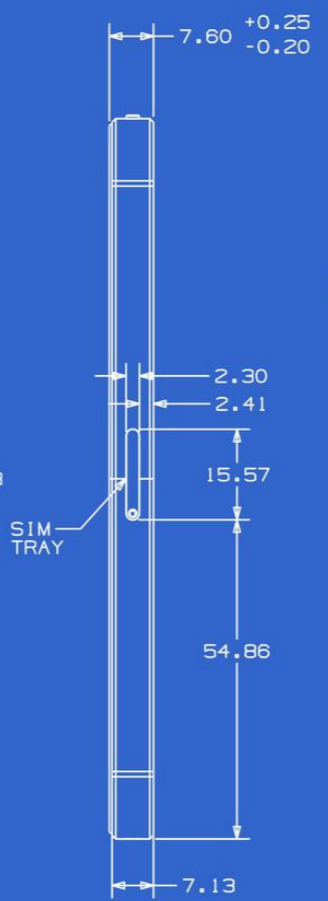
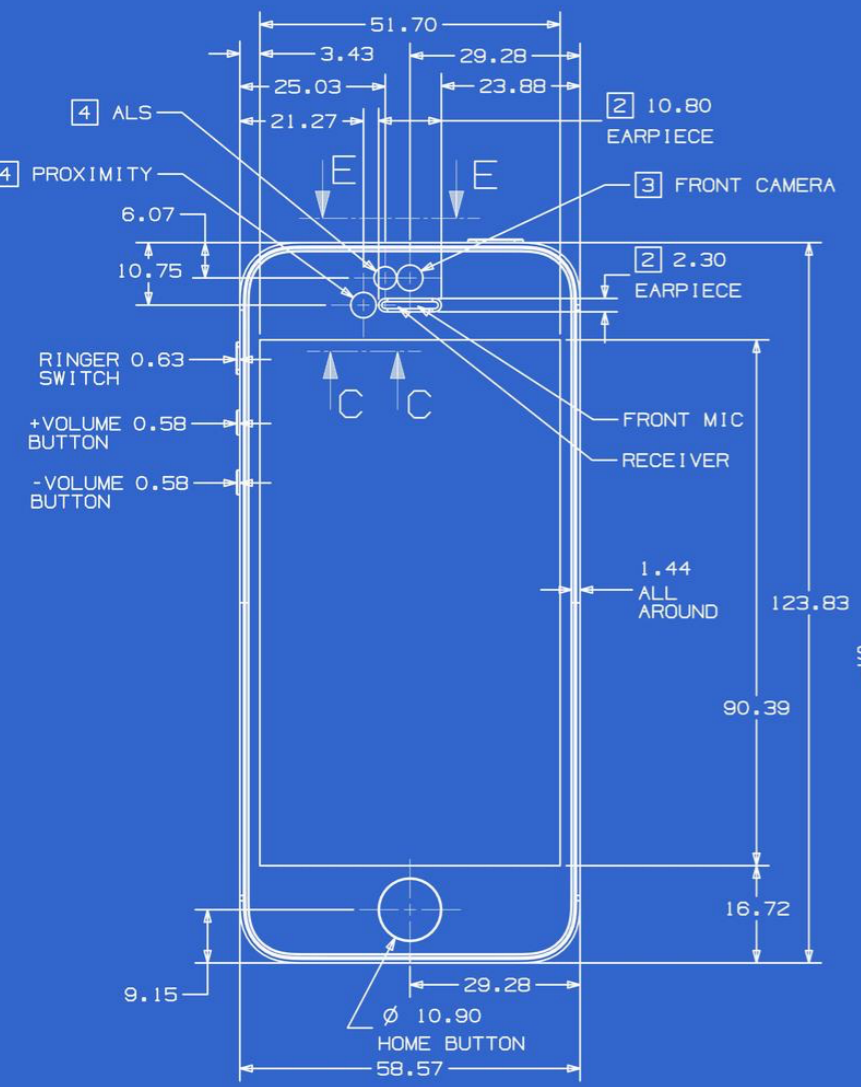
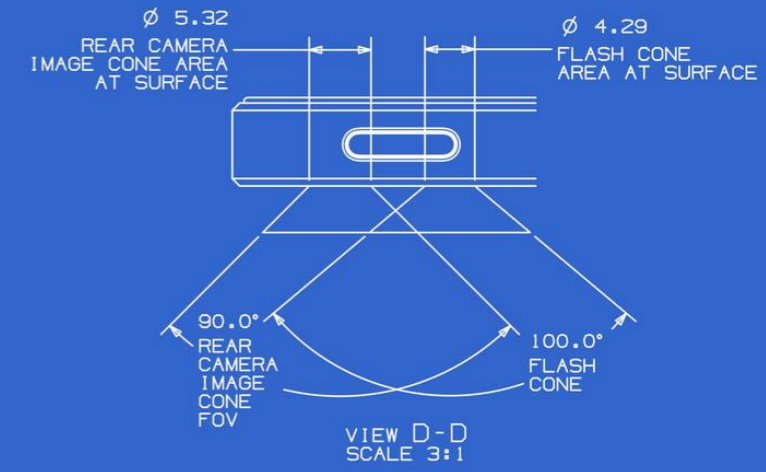
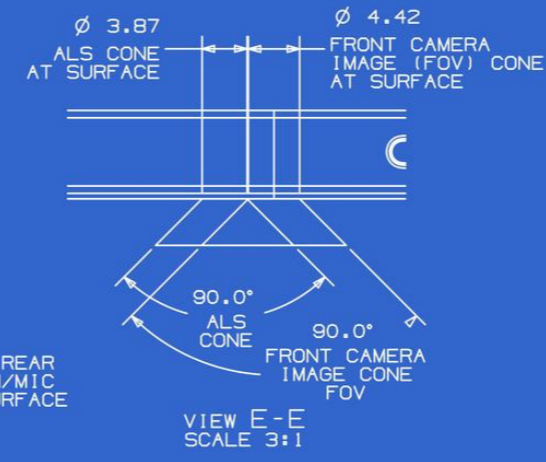
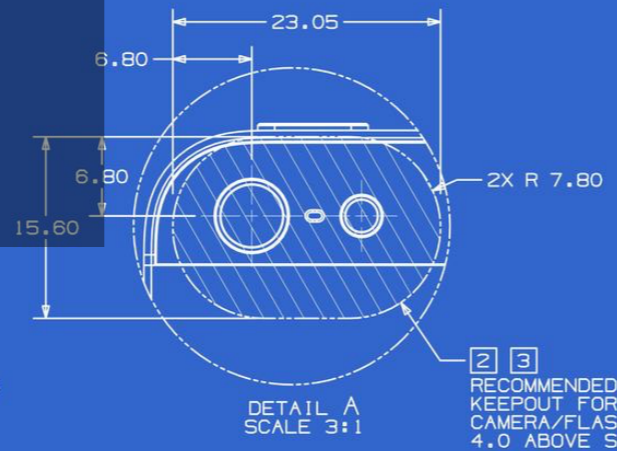
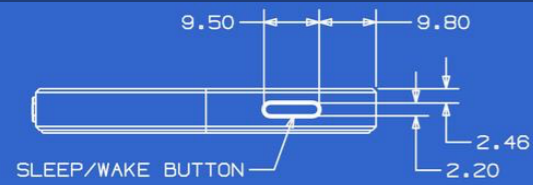


Quality is relative



Standard

5 METAL.
 OPENINGS: FRONT MIC, REAR MIC, EARPIECE, AND SPEAKER.
 FEATURES: FRONT CAMERA, REAR CAMERA, REAR FLASH,
 SENSOR OR ALS (AMBIENT LIGHT SENSOR).

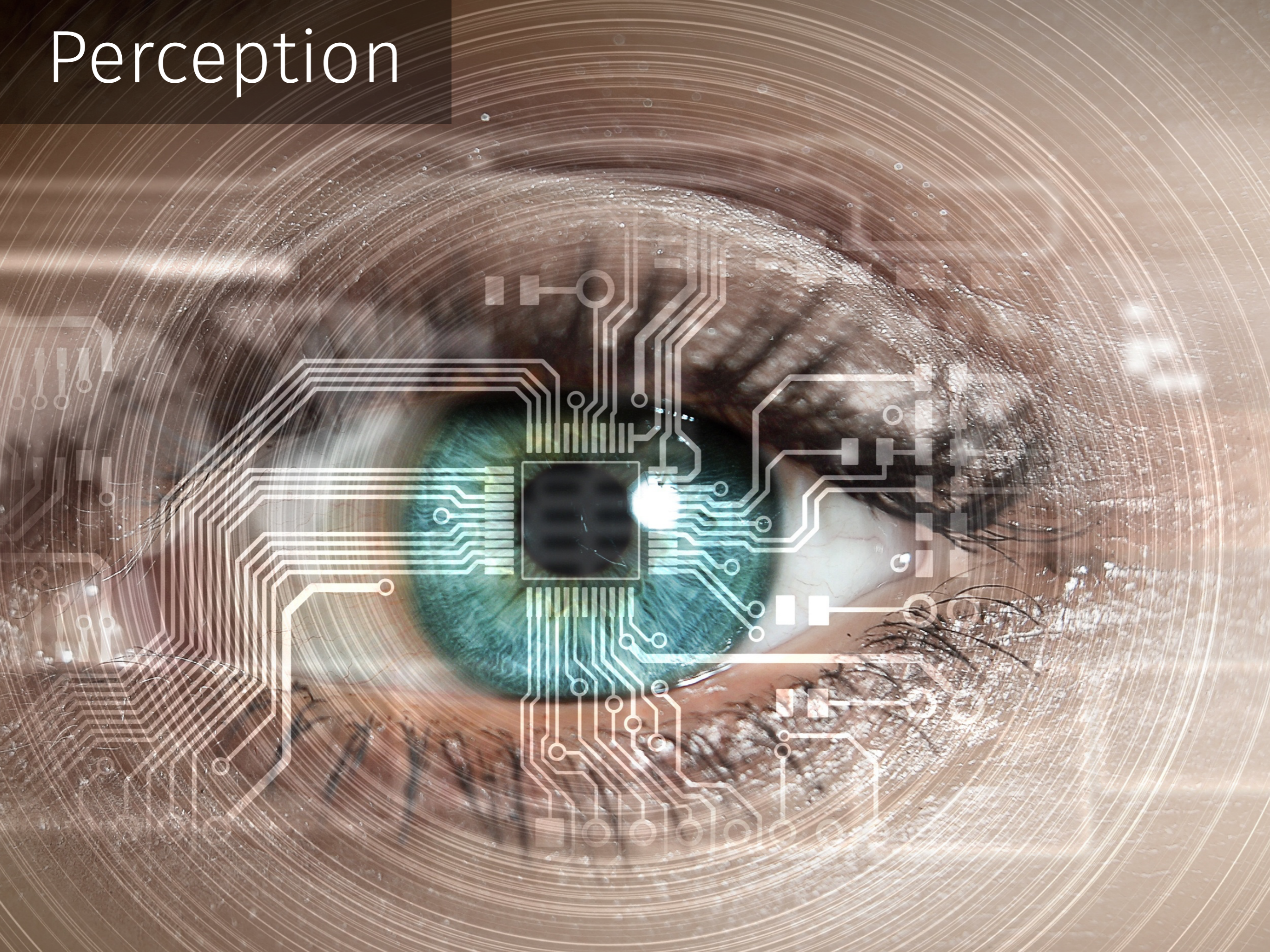


METRIC		Apple Inc.	
DRAFTER APPLE INC.	DATE 09/12/12	NOTICE OF PROPRIETARY PROPERTY: THE INFORMATION CONTAINED HEREIN IS THE PROPRIETARY PROPERTY OF APPLE INC.	
DESIGNER APPLE INC.	DATE 09/12/12	©2010 APPLE INC. ALL RIGHTS RESERVED. APPLE, THE APPLE LOGO, AND iPhone ARE TRADEMARKS OF APPLE INC. REGISTERED IN THE U.S. AND OTHER COUNTRIES.	
DIMENSIONS ARE IN MILLIMETERS		TITLE IPHONE 5	
TOLERANCES		DRAWING NUMBER	
X.X	±0.4	REV.	
X.XX	±0.20		
X.XXX	±0.100		
ANGLES ±0.5°			
DO NOT SCALE DRAWINGS			
THIRD ANGLE PROJECTION	SIZE D	SCALE NONE	SHT 1 OF 1

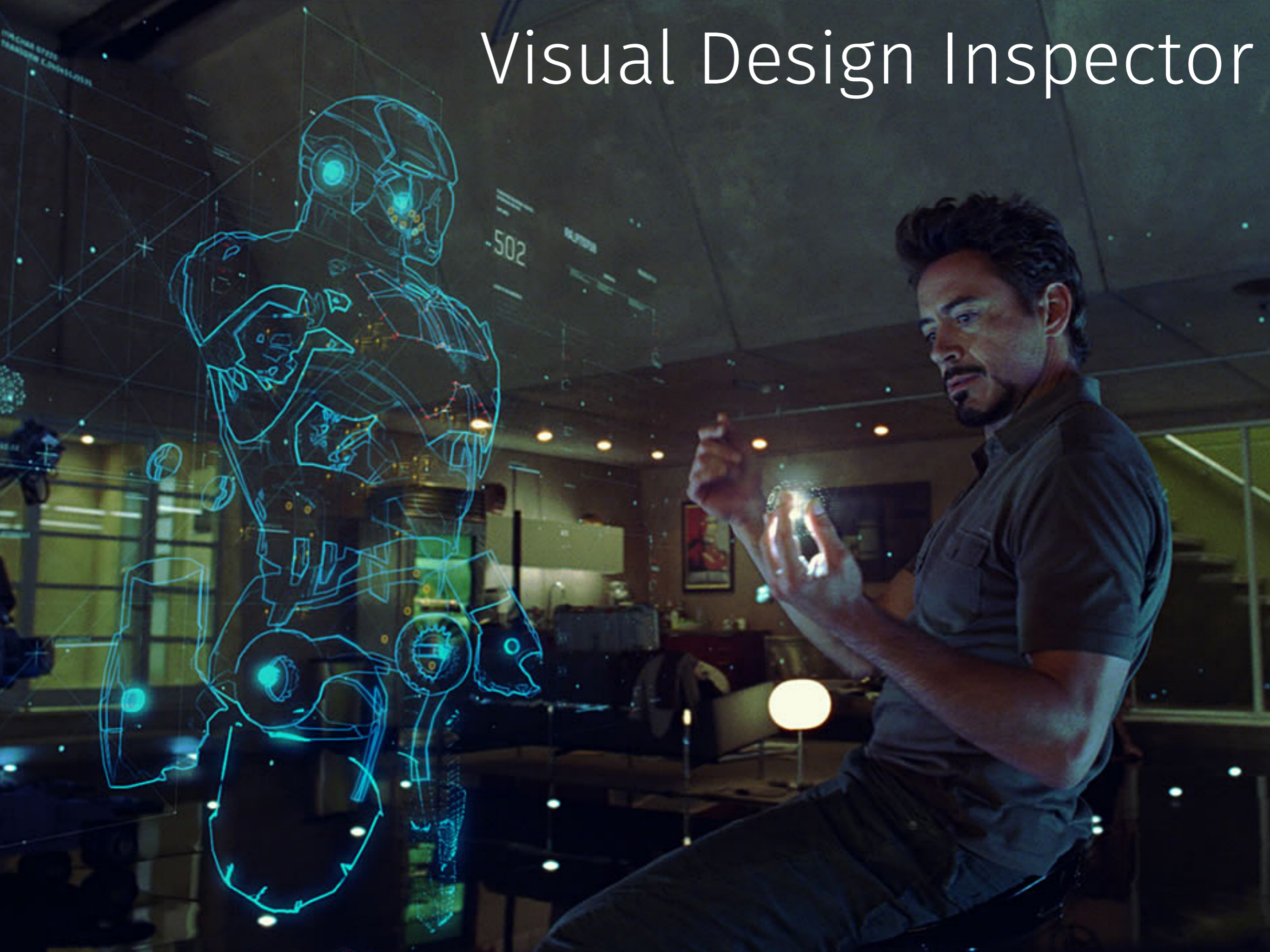
Comparison



Perception



Visual Design Inspector





Select packages

DevFlow-UI (12c 182m)
DevFlow-Uploader (3c 12m)
DevFlow-Utils (62c 586m)
EmbeddedFreeType (6c 47m)
EmbeddedFreeType-Tests (2c 5m)
EmergencyEvaluator (1c 22m)
Epicea (128c 1,170m)
FileSystem-Core (38c 658m)
FileSystem-Disk (9c 113m)
FileSystem-Memory (8c 92m)
FileSystem-Tests-Core (21c 305m)
FileSystem-Tests-Disk (3c 10m)
FileSystem-Tests-Memory (2c 4m)
FileSystem-Zip (4c 15m)
Files (21c 349m)
FontChooser (4c 120m)
FontInfrastructure (5c 151m)
FontInfrastructure-Tests (2c 3m)
FreeType (38c 529m)
FreeTypeTests (1c 26m)
Fuel (125c 583m)
FuelCommandLineHandler (1c 8m)
FuelSystem-FileRegistry (1c 2m)
FuelTests (55c 365m)
FuelTools-Debugger (1c 8m)
GT-Inspector (15c 155m)
GT-InspectorExtensions-Core (147c 258m)
GT-Playground (20c 154m)
GT-Spotter (84c 928m)
GT-Spotter-EventRecorder (17c 228m)
GT-SpotterExtensions-Core (57c 166m)

DevFlow-Profilers (24c 119m)

Add

Add All

Remove

Remove All

Next

Select packages

Vidi

Select All Deselect All

Rule	Critics
Style	63
Design Flaws	77
Class not referenced	5
Excessive inheritance depth	0
Excessive number of arguments	0
Excessive number of methods	2
Excessive number of variables	3
Future deprecation warning	2
Inconsistent method classificatio	58
Instance variables defined in all s	0
Long methods	4
Method defined in all subclasses,	0
Methods equivalently defined in :	0
Refers to class name instead of "	0
Rewrite super messages to self n	1
Sends a deprecated message to a	0
Sends different super message	2
Variable is only assigned a single	0
Variable referenced in only one n	0
Variables not referenced	0
Unclassified rules	0
Coding Idiom Violation	23
Bugs	6
Potential Bugs	6
Optimization	7

City

DFSingletonWindows class>>#new | 2 violations

DFSessionsBrowserWindow>>#openTopRightWithSpec

Violated rules:

- Rewrite super messages to self messages when both refer
- Sends different super message

Source code:

```

openTopRightWithSpec

| w topRight |

"If it is open, do not open"
self class isOpen ifTrue: [ ^ nil ].

topRight := (World fullBounds width - 15 - self
initialExtent x) @ 15.

w := super openAtWithSpec: topRight.

w whenClosedDo: [
    DFSessionsManager uniqueInstance announcer
unsubscribe: self.
self class reset.
].

```

Total classes: 36 methods: 299

Violated classes: 12 methods: 137; with a total number of violations: 182

Rule	Critics
<input checked="" type="checkbox"/> Style	63
<input checked="" type="checkbox"/> Design Flaws	77
<input checked="" type="checkbox"/> Class not referenced	5
<input checked="" type="checkbox"/> Excessive inheritance depth	0
<input checked="" type="checkbox"/> Excessive number of arguments	0
<input checked="" type="checkbox"/> Excessive number of methods	2
<input checked="" type="checkbox"/> Excessive number of variables	3
<input checked="" type="checkbox"/> Future deprecation warning	2
<input checked="" type="checkbox"/> Inconsistent method classificatio	58
<input checked="" type="checkbox"/> Instance variables defined in all s	0
<input checked="" type="checkbox"/> Long methods	4
<input checked="" type="checkbox"/> Method defined in all subclasses,	0
<input checked="" type="checkbox"/> Methods equivalently defined in :	0
<input checked="" type="checkbox"/> Refers to class name instead of "	0
<input checked="" type="checkbox"/> Rewrite super messages to self n	1
<input checked="" type="checkbox"/> Sends a deprecated message to a	0
<input checked="" type="checkbox"/> Sends different super message	2
<input checked="" type="checkbox"/> Variable is only assigned a single	0
<input checked="" type="checkbox"/> Variable referenced in only one n	0
<input checked="" type="checkbox"/> Variables not referenced	0
<input checked="" type="checkbox"/> Unclassified rules	0
<input checked="" type="checkbox"/> Coding Idiom Violation	23
<input checked="" type="checkbox"/> Bugs	6
<input checked="" type="checkbox"/> Potential Bugs	6
<input checked="" type="checkbox"/> Optimization	7

Vidi

City

DFSingletonWindows class>>#new | 2 violations

Total classes: 36 methods: 299
 Violated classes: 12 methods: 137; with a total number of violations: 182

DFSessionsBrowserWindow>>#openTopRightWithSpec

Violated rules:

- Rewrite super messages to self messages when both refer
- Sends different super message

Source code:

```

openTopRightWithSpec

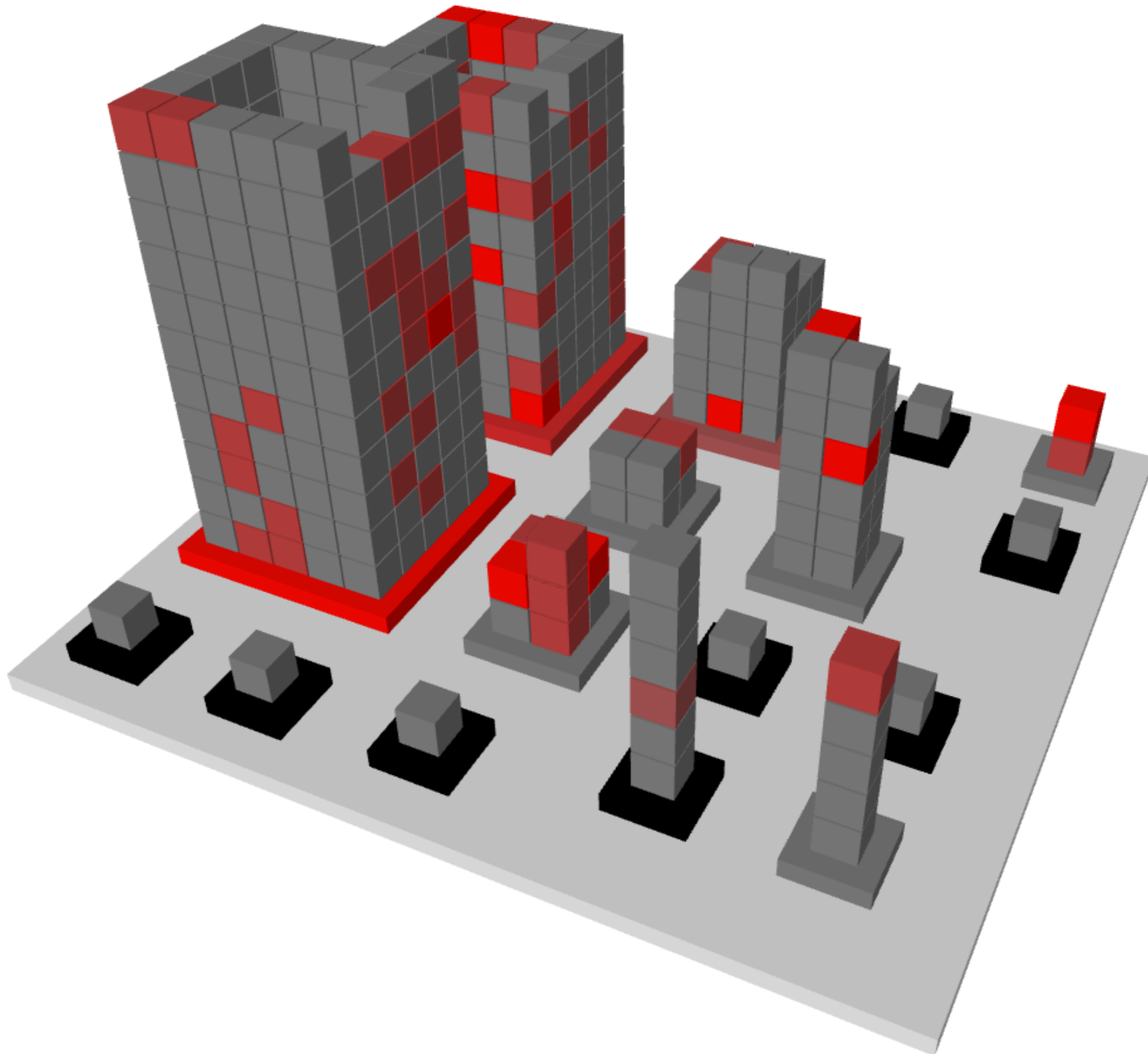
| w topRight |

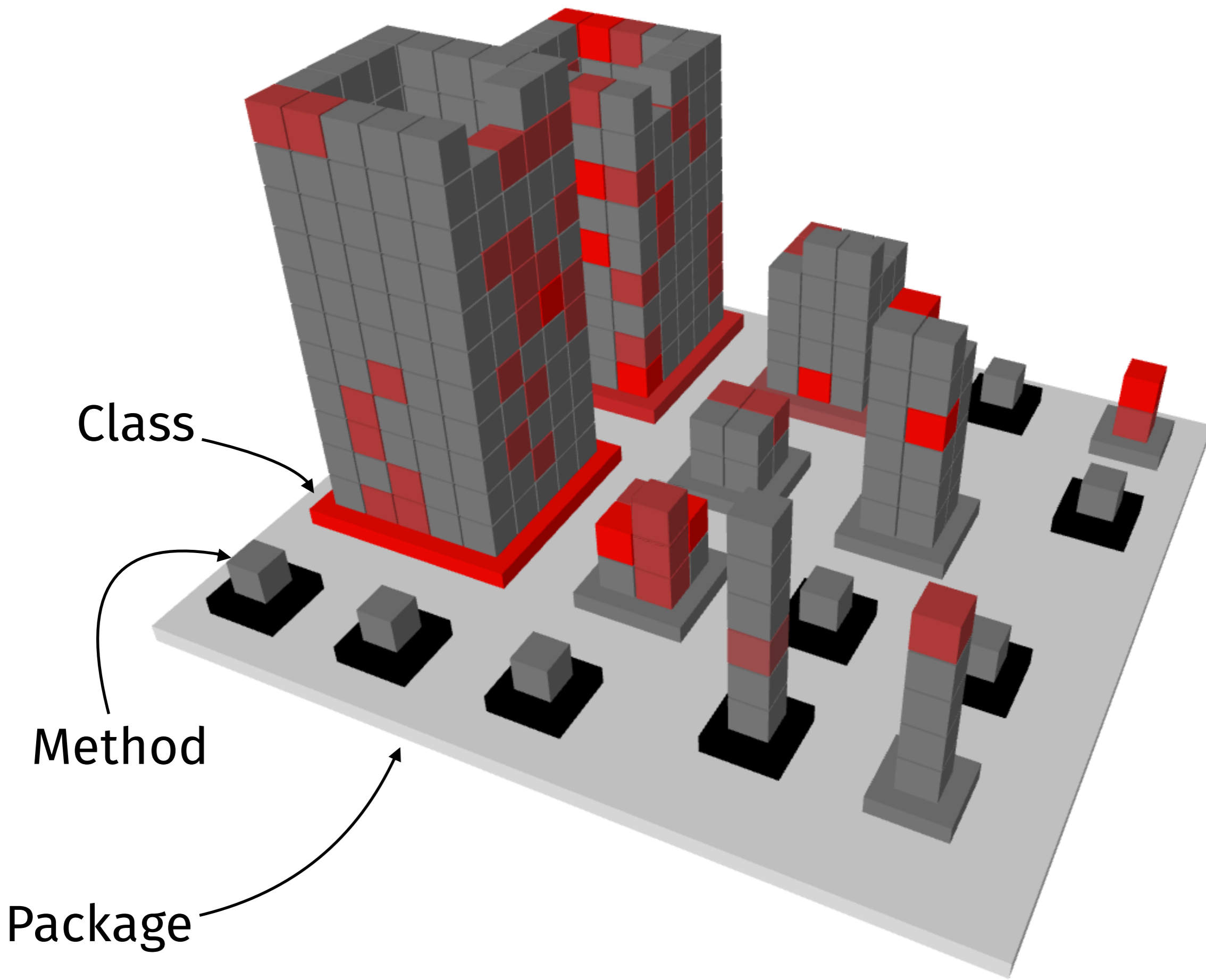
"If it is open, do not open"
self class isOpen ifTrue: [ ^ nil ].

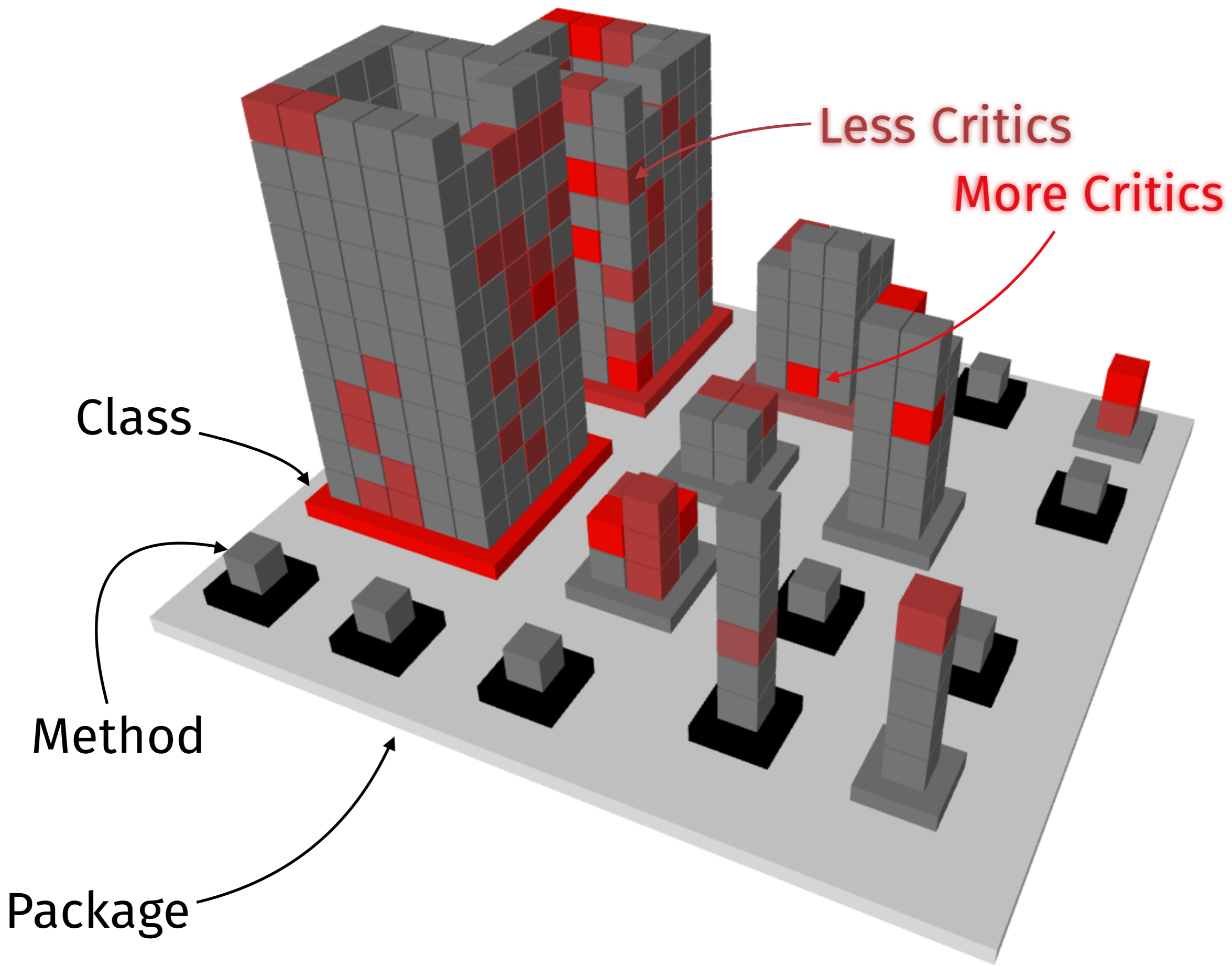
topRight := (World fullBounds width - 15 - self
initialExtent x) @ 15.

w := super openAtWithSpec: topRight.

w whenClosedDo: [
  DFSessionsManager uniqueInstance announcer
unsubscribe: self.
self class reset.
].
  
```







Class

Method

Package

Less Critics

More Critics

Rule	Critics
<input checked="" type="checkbox"/> Style	63
<input checked="" type="checkbox"/> Design Flaws	77
<input checked="" type="checkbox"/> Class not referenced	5
<input checked="" type="checkbox"/> Excessive inheritance depth	0
<input checked="" type="checkbox"/> Excessive number of arguments	0
<input checked="" type="checkbox"/> Excessive number of methods	2
<input checked="" type="checkbox"/> Excessive number of variables	3
<input checked="" type="checkbox"/> Future deprecation warning	2
<input checked="" type="checkbox"/> Inconsistent method classificatio	58
<input checked="" type="checkbox"/> Instance variables defined in all s	0
<input checked="" type="checkbox"/> Long methods	4
<input checked="" type="checkbox"/> Method defined in all subclasses,	0
<input checked="" type="checkbox"/> Methods equivalently defined in :	0
<input checked="" type="checkbox"/> Refers to class name instead of "	0
<input checked="" type="checkbox"/> Rewrite super messages to self n	1
<input checked="" type="checkbox"/> Sends a deprecated message to a	0
<input checked="" type="checkbox"/> Sends different super message	2
<input checked="" type="checkbox"/> Variable is only assigned a single	0
<input checked="" type="checkbox"/> Variable referenced in only one n	0
<input checked="" type="checkbox"/> Variables not referenced	0
<input checked="" type="checkbox"/> Unclassified rules	0
<input checked="" type="checkbox"/> Coding Idiom Violation	23
<input checked="" type="checkbox"/> Bugs	6
<input checked="" type="checkbox"/> Potential Bugs	6
<input checked="" type="checkbox"/> Optimization	7

Vidi

City

DFSingletonWindows class>>#new | 2 violations

Total classes: 36 methods: 299
 Violated classes: 12 methods: 137; with a total number of violations: 182

DFSessionsBrowserWindow>>#openTopRightWithSpec

Violated rules:

- Rewrite super messages to self messages when both refer
- Sends different super message

Source code:

```

openTopRightWithSpec

| w topRight |

"If it is open, do not open"
self class isOpen ifTrue: [ ^ nil ].

topRight := (World fullBounds width - 15 - self
initialExtent x) @ 15.

w := super openAtWithSpec: topRight.

w whenClosedDo: [
  DFSessionsManager uniqueInstance announcer
unsubscribe: self.
self class reset.
].
  
```

Rule	Critics
<input checked="" type="checkbox"/> Style	63
<input checked="" type="checkbox"/> Design Flaws	77
<input checked="" type="checkbox"/> Class not referenced	5
<input checked="" type="checkbox"/> Excessive inheritance depth	0
<input checked="" type="checkbox"/> Excessive number of arguments	0
<input checked="" type="checkbox"/> Excessive number of methods	2
<input checked="" type="checkbox"/> Excessive number of variables	3
<input checked="" type="checkbox"/> Future deprecation warning	2
<input checked="" type="checkbox"/> Inconsistent method classificatio	58
<input checked="" type="checkbox"/> Instance variables defined in all s	0
<input checked="" type="checkbox"/> Long methods	4
<input checked="" type="checkbox"/> Method defined in all subclasses,	0
<input checked="" type="checkbox"/> Methods equivalently defined in :	0
<input checked="" type="checkbox"/> Refers to class name instead of "	0
<input checked="" type="checkbox"/> Rewrite super messages to self n	1
<input checked="" type="checkbox"/> Sends a deprecated message to a	0
<input checked="" type="checkbox"/> Sends different super message	2
<input checked="" type="checkbox"/> Variable is only assigned a single	0
<input checked="" type="checkbox"/> Variable referenced in only one n	0
<input checked="" type="checkbox"/> Variables not referenced	0
<input checked="" type="checkbox"/> Unclassified rules	0
<input checked="" type="checkbox"/> Coding Idiom Violation	23
<input checked="" type="checkbox"/> Bugs	6
<input checked="" type="checkbox"/> Potential Bugs	6
<input checked="" type="checkbox"/> Optimization	7

Vidi

City

DFSingletonWindows class>>#new | 2 violations

Total classes: 36 methods: 299
 Violated classes: 12 methods: 137; with a total number of violations: 182

DFSessionsBrowserWindow>>#openTopRightWithSpec

Violated rules:

- Rewrite super messages to self messages when both refer
- Sends different super message

Source code:

```

openTopRightWithSpec

| w topRight |

"If it is open, do not open"
self class isOpen ifTrue: [ ^ nil ].

topRight := (World fullBounds width - 15 - self
initialExtent x) @ 15.

w := super openAtWithSpec: topRight.

w whenClosedDo: [
  DFSessionsManager uniqueInstance announcer
unsubscribe: self.
self class reset.
].
  
```

Vidi

City

Total classes: 36 methods: 299
Violated classes: 12 methods: 137; with a total number of violations: 182

Select All Deselect All

Rule	Critics
<input checked="" type="checkbox"/> Style	63
<input checked="" type="checkbox"/> Design Flaws	77
<input checked="" type="checkbox"/> Class not referenced	5
<input checked="" type="checkbox"/> Excessive inheritance depth	0
<input checked="" type="checkbox"/> Excessive number of arguments	0
<input checked="" type="checkbox"/> Excessive number of methods	2
<input checked="" type="checkbox"/> Excessive number of variables	3
<input checked="" type="checkbox"/> Future deprecation warning	2
<input checked="" type="checkbox"/> Inconsistent method classificatio	58
<input checked="" type="checkbox"/> Instance variables defined in all s	0
<input checked="" type="checkbox"/> Long methods	4
<input checked="" type="checkbox"/> Method defined in all subclasses,	0
<input checked="" type="checkbox"/> Methods equivalently defined in :	0
<input checked="" type="checkbox"/> Refers to class name instead of "	0
<input checked="" type="checkbox"/> Rewrite super messages to self n	1
<input checked="" type="checkbox"/> Sends a deprecated message to a	0
<input checked="" type="checkbox"/> Sends different super message	2
<input checked="" type="checkbox"/> Variable is only assigned a single	0
<input checked="" type="checkbox"/> Variable referenced in only one n	0
<input checked="" type="checkbox"/> Variables not referenced	0
<input checked="" type="checkbox"/> Unclassified rules	0
<input checked="" type="checkbox"/> Coding Idiom Violation	23
<input checked="" type="checkbox"/> Bugs	6
<input checked="" type="checkbox"/> Potential Bugs	6
<input checked="" type="checkbox"/> Optimization	7

DFSessionsBrowserWindow>>#openTopRightWithSpec

Violated rules:

- Rewrite super messages to self messages when both refer
- Sends different super message**

Source code:

```

openTopRightWithSpec

| w topRight |

"If it is open, do not open"
self class isOpen ifTrue: [ ^ nil ].

topRight := (World fullBounds width - 15 - self
initialExtent x) @ 15.

w := super openAtWithSpec: topRight.

w whenClosedDo: [
    DFSessionsManager uniqueInstance announcer
unsubscribe: self.
self class reset.
].

```

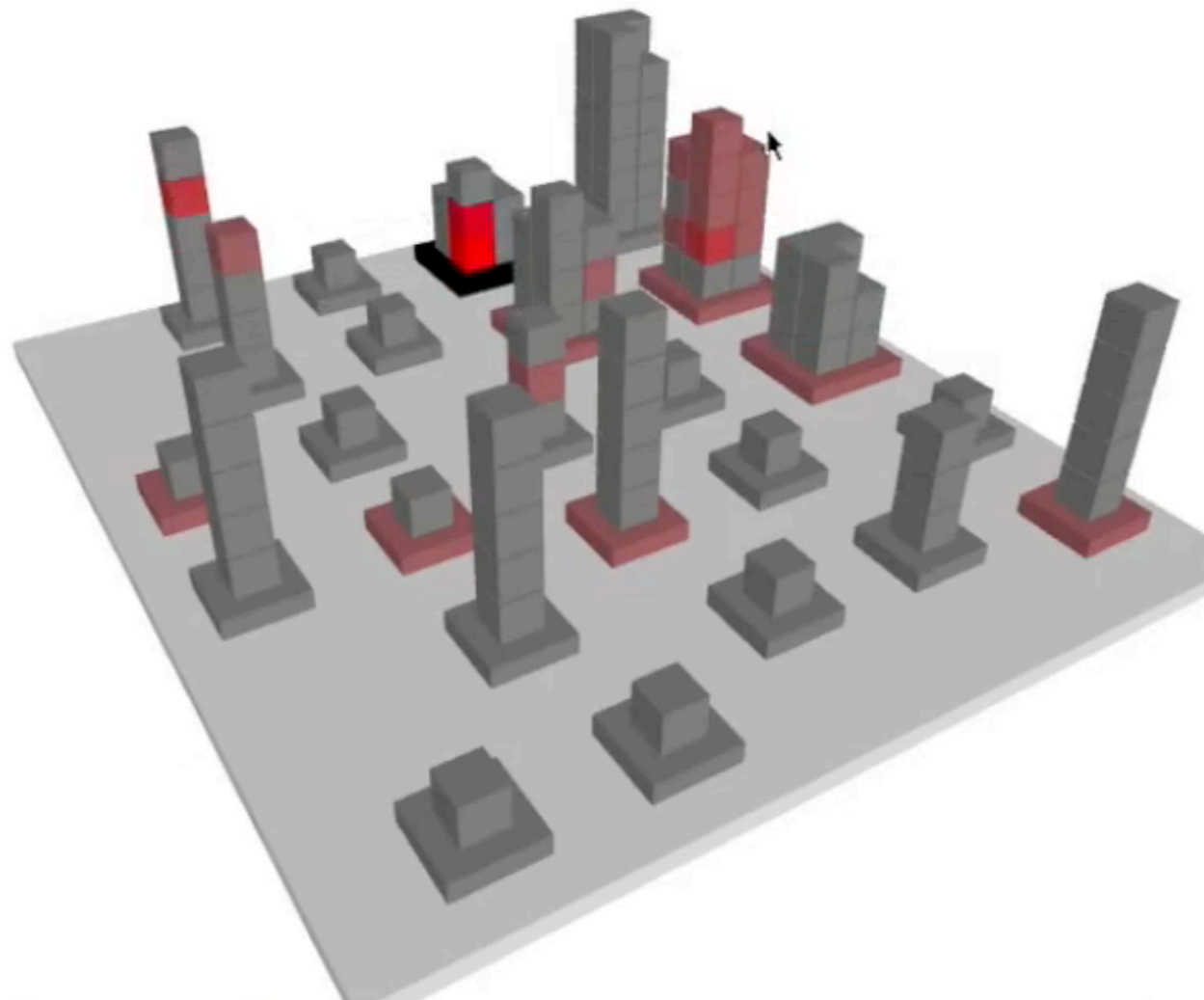



Vidi

Select All Deselect All

City

Rule	Critic
<input checked="" type="checkbox"/> Optimization	3
<input checked="" type="checkbox"/> Design Flaws	33
<input checked="" type="checkbox"/> Class not referenc	4
<input checked="" type="checkbox"/> Excessive inheritar	0
<input checked="" type="checkbox"/> Excessive number	0
<input checked="" type="checkbox"/> Excessive number	0
<input checked="" type="checkbox"/> Excessive number	0
<input checked="" type="checkbox"/> Future deprecation	2
<input type="checkbox"/> Inconsistent metho	23
<input checked="" type="checkbox"/> Instance variables	0
<input checked="" type="checkbox"/> Long methods	2
<input checked="" type="checkbox"/> Method defined in	0
<input checked="" type="checkbox"/> Methods equivalen	0
<input checked="" type="checkbox"/> Refers to class nan	2
<input checked="" type="checkbox"/> Rewrite super mes	0
<input checked="" type="checkbox"/> Sends a deprecate	0
<input checked="" type="checkbox"/> Sends different su	0
<input checked="" type="checkbox"/> Variable is only as:	0
<input checked="" type="checkbox"/> Variable reference	0
<input checked="" type="checkbox"/> Variables not refer	0
<input checked="" type="checkbox"/> Coding Idiom Violati	15
<input checked="" type="checkbox"/> Unclassified rules	0
<input checked="" type="checkbox"/> Bugs	2
<input checked="" type="checkbox"/> Potential Bugs	3
<input type="checkbox"/> Style	11



Total classes: 24 methods: 119
Violated classes: 8 methods: 70; with a total number of violations: 100

Violated rules:

Source code:

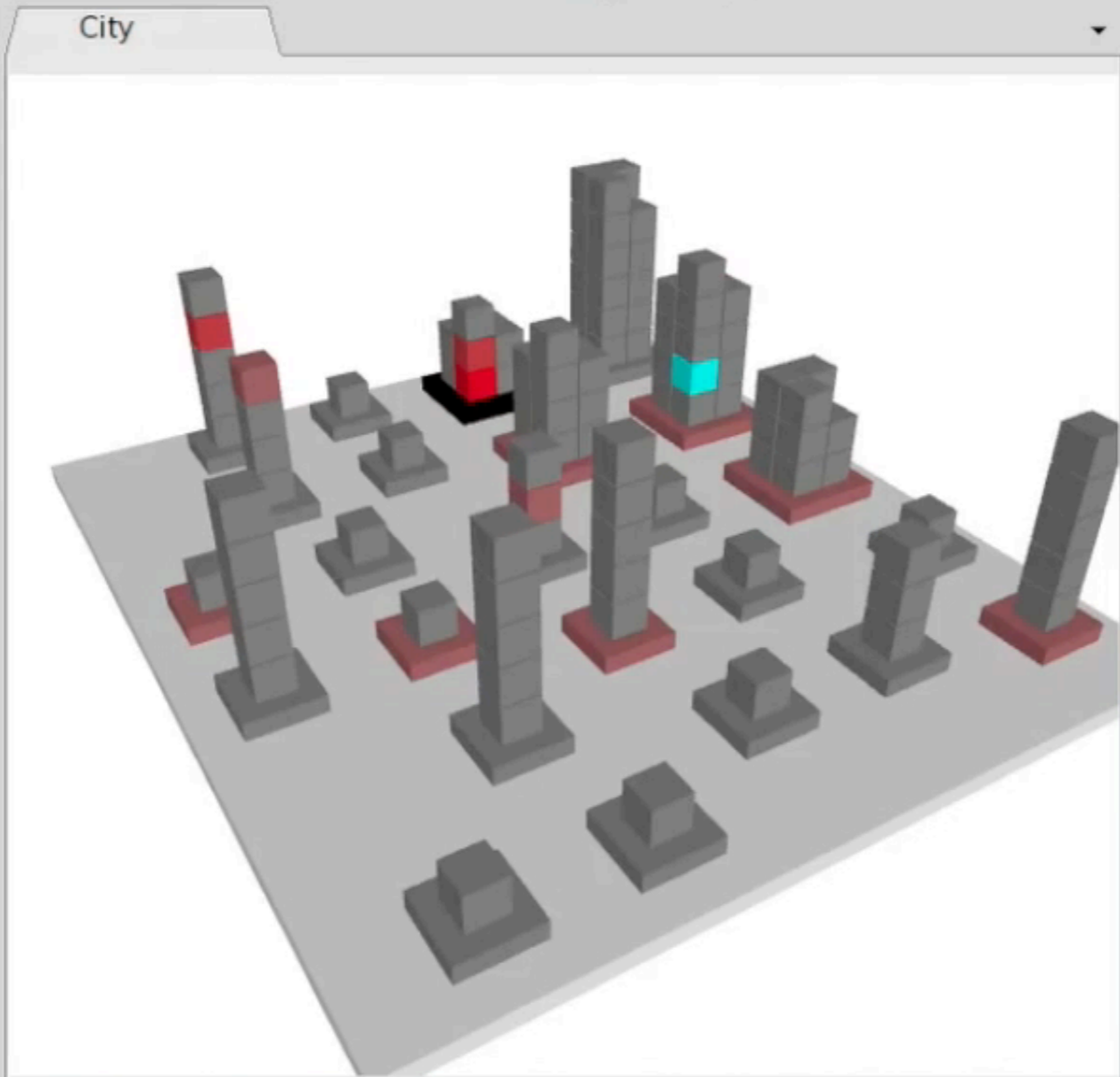
Vidi

Revalidating DFWindowsProfiler class>>#triggerEvent:forSystemWindow:

Vidi

Select All Deselect All

Rule	Critic
▶ Optimization	3
▶ Design Flaws	33
▶ Coding Idiom Violation	14
▶ Unclassified rules	0
▶ Bugs	2
▶ Potential Bugs	3
▶ Style	44



Total classes: 24 methods: 119
Violated classes: 8 methods: 69; with a total number of violations: 99

DFWindowsProfiler class>>#triggerEvent:forSystemWindow:

- Violated rules:
- Eliminate unnecessary not's
 - Eliminate guarding clauses**

Source code:

```
    | profiledWindow event |  
  
    profiledWindow := DFProfiledWindow  
fromSystemWindow: aSystemWindow.  
  
    (anEventClass = DFCloseWindowEvent)  
    ifTrue: [ profiledWindow closedNow ].  
  
    profiledWindow isNil ifFalse: [  
        event := anEventClass for: profiledWindow id.  
        self updateWindow: profiledWindow  
withEvent: event.
```

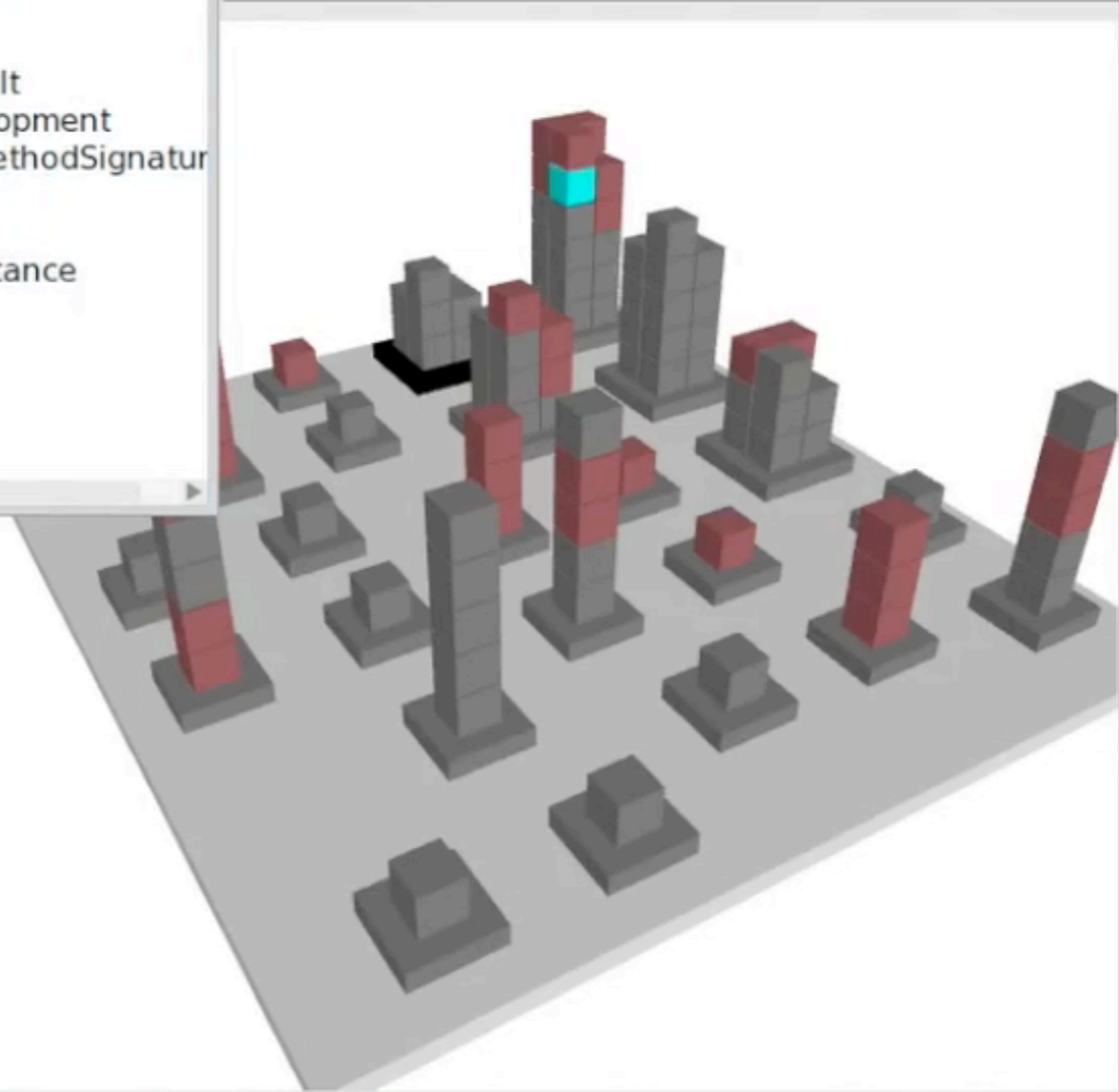
Vidi



Vidi

DFProfilersSuite class

- all --
- as yet unclassified
- instance c
- #uniqueInstance
- install
- new
- newDefault
- newDevelopment
- profiledMethodSignatur
- reset
- uninstall
- uniqueInstance



Total classes: 24 methods: 119
 Violated classes: 8 methods: 46; with a total number of violations: 64

DFProfilersSuite class>>#uniqueInstance

Violated rules:
 Unclassified methods

Source code:

```
uniqueInstance

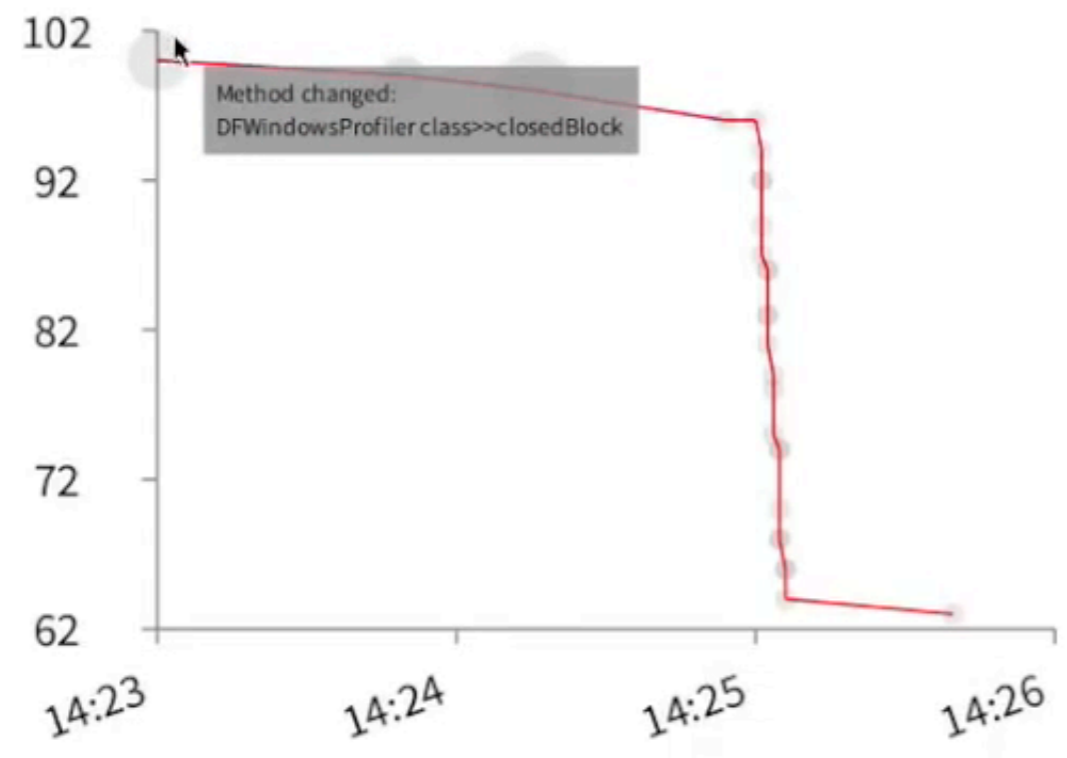
UniqueInstance ifNil: [ UniqueInstance := self
newDefault ].
^UniqueInstance
```



Inspector on a VidiSessionManager class (VidiSessionManager)

a VidiSession
Raw Quality Timeline Meta

Enlarge points Reduce points



a VidiSessionEvent x
Diff Raw Meta

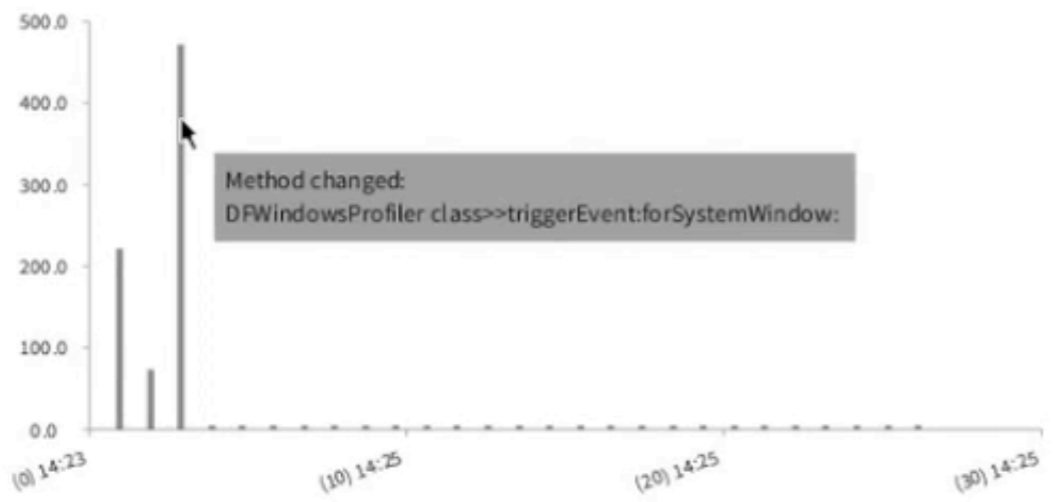
```
triggerEvent: anEventClass forSystem
| profiledWindow event |
profiledWindow := DFProfiledWindow
(anEventClass = DFCloseWindowEvent
ifTrue: [ profiledWindow closedBlock
profiledWindow isNil ifFalse: [
event := anEventClass for: profiledWindow
self updateWindow: profiledWindow
].
profiledWindow := DFProfiledWindow
ifTrue: [ profiledWindow closedBlock
profiledWindow isNil
ifTrue: [ ^ self ].
event := anEventClass for: profiledWindow
self updateWindow: profiledWindow
].
profiledWindow := DFProfiledWindow
ifTrue: [ profiledWindow closedBlock
profiledWindow isNil
ifTrue: [ ^ self ].
event := anEventClass for: profiledWindow
self updateWindow: profiledWindow
].
profiledWindow := DFProfiledWindow
ifTrue: [ profiledWindow closedBlock
profiledWindow isNil
ifTrue: [ ^ self ].
event := anEventClass for: profiledWindow
self updateWindow: profiledWindow
].
profiledWindow := DFProfiledWindow
ifTrue: [ profiledWindow closedBlock
profiledWindow isNil
ifTrue: [ ^ self ].
event := anEventClass for: profiledWindow
self updateWindow: profiledWindow
].
```




Inspector on a VidiSessionManager class (VidiSessionManager)

a VidiSession
Raw Quality Timeline Meta

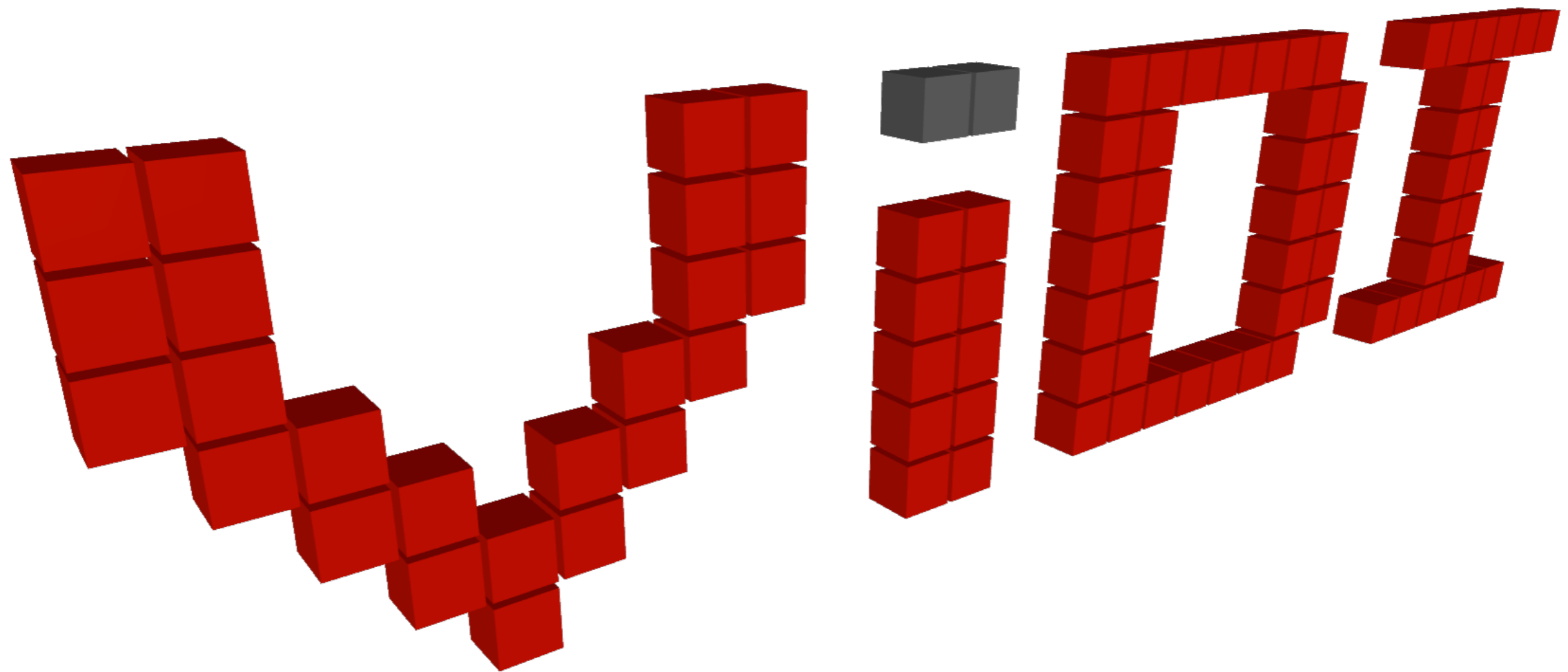
a VidiSessionEvent x
Diff Raw Meta



```
triggerEvent: anEventClass forSystemWindow: [ profiledWindow event ]
| profiledWindow event |
profiledWindow := DFProfiledWindow new
(anEventClass = DFCloseWindowEvent)
ifTrue: [ profiledWindow closed ]

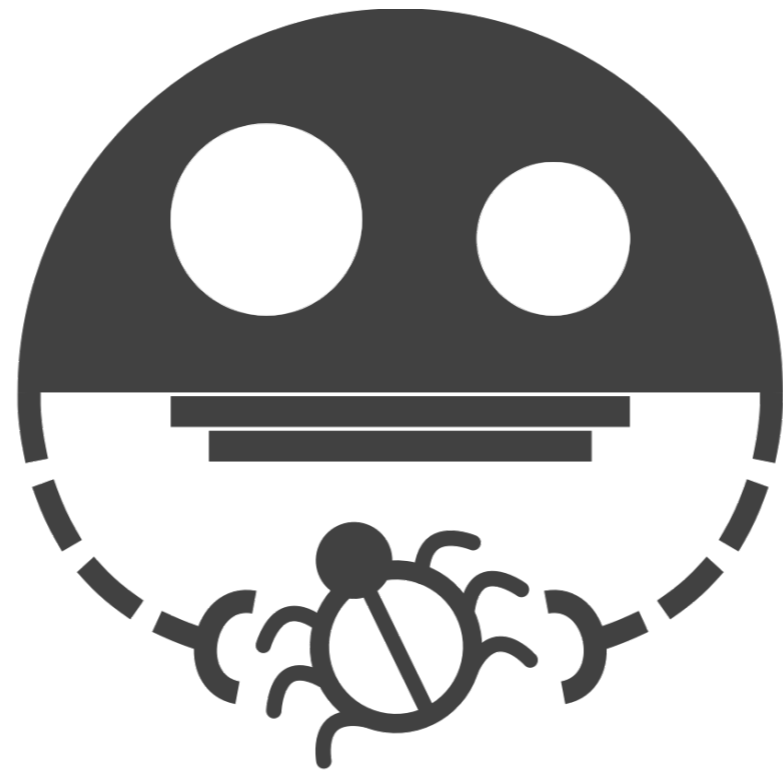
profiledWindow isNil ifFalse: [
  event := anEventClass for: profiledWindow
  self updateWindow: profiledWindow
].

triggerEvent: anEventClass forSystemWindow: [ profiledWindow event ]
profiledWindow := DFProfiledWindow new
anEventClass = DFCloseWindowEvent
ifTrue: [ profiledWindow closed ]
profiledWindow isNil
ifTrue: [ ^ self ].
event := anEventClass for: profiledWindow
self updateWindow: profiledWindow
```

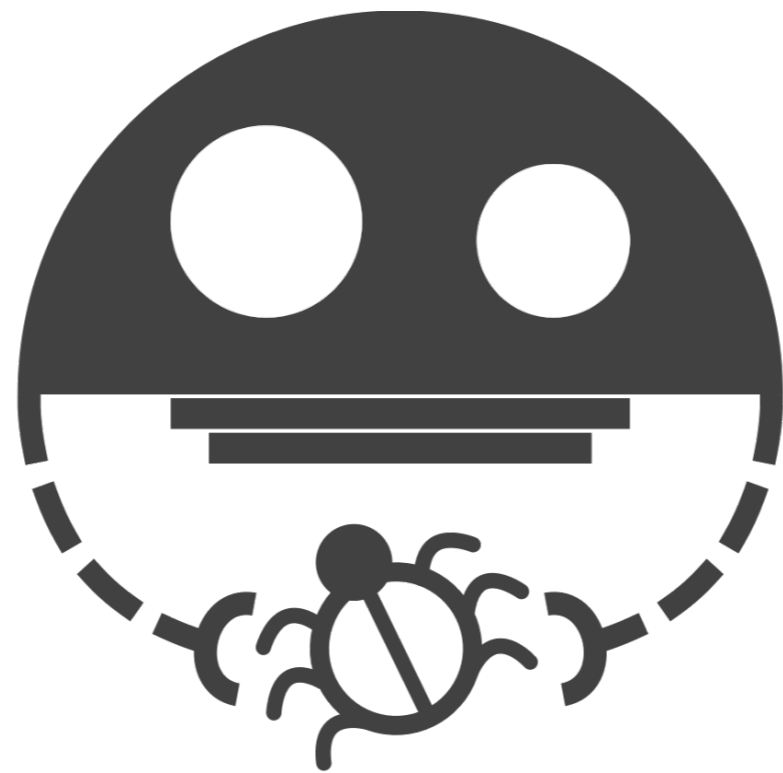


<http://yuriy.tymch.uk/Vidi/>

QualityAssistant



QualityAssistant



1.0

Quality Assurance Engineer





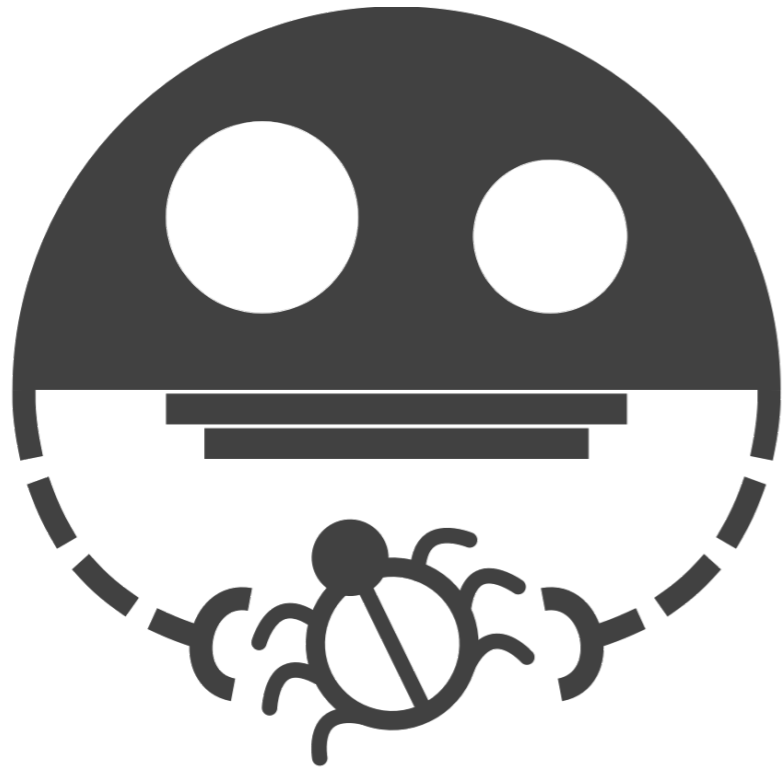
Quality Assurance Engineer

“Our QA is carrying his son”

“Our QA has a birthday today”

“The QA will check my patch”

Quality Assistant

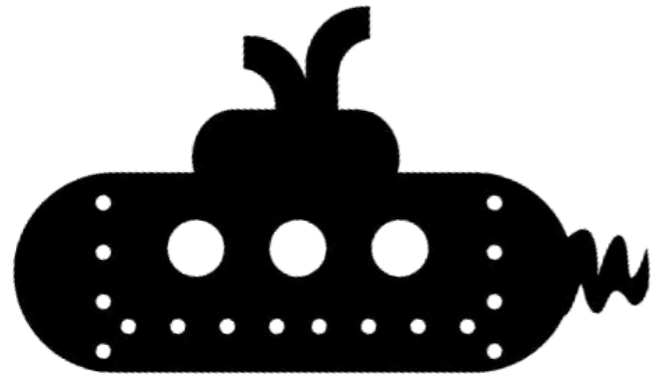


`QA criticsOf: aClass`

`QA revalidate: aMethod`



Quality Assistant



Nautilus



Inspector



Spotter



Quality Assistant

```
exportSettingAction
| targetSymbol currentValue defaultValue |

targetSymbol := self target isSymbol
  ifTrue: [ self target ]
  ifFalse: [ self target name asSymbol ].

currentValue := self targetSelector
  ifNil: [ (Smalltalk at: targetSymbol)
    perform: self getSelector ]
  ifNotNil: [:s |
    ((Smalltalk at: targetSymbol)
    perform: s) perform: self getSelector].

^ (self hasDefault not or: [
  (self default = currentValue) not ])
  ifTrue: [ self startupAction ]
  ifFalse: [ "We do nothing" nil ]
```



{i}

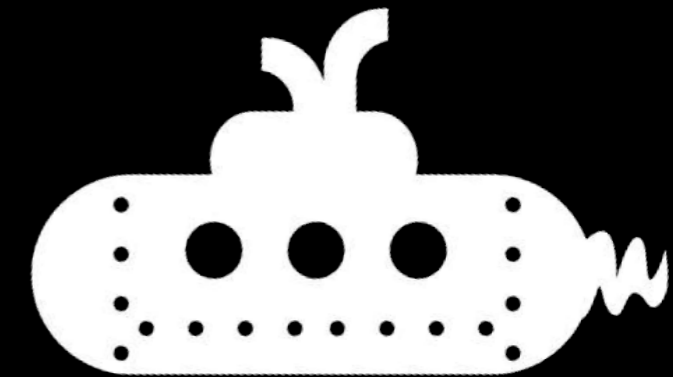
{c}

371

⚠ Eliminate unnecessary not's ? ✖ This is useless!

❗ Messages sent but not implemented ? ✖ This is useless!

ℹ Temporary variables not read AND written ? ✖ This is useless!



Nautilus



Quality Assistant

```
exportSettingAction
| targetSymbol currentValue defaultValue |

targetSymbol := self target isSymbol
  ifTrue: [ self target ]
  ifFalse: [ self target name asSymbol ].




currentValue := self targetSelector
  ifNil: [ (Smalltalk at: targetSymbol)
    perform: self getSelector ]
  ifNotNil: [:s |
    ((Smalltalk at: targetSymbol)
    perform: s) perform: self getSelector].

^ (self hasDefault not or: [
  (self default = currentValue) not ])
  ifTrue: [ self startupAction ]
  ifFalse: [ "We do nothing" nil ]
```



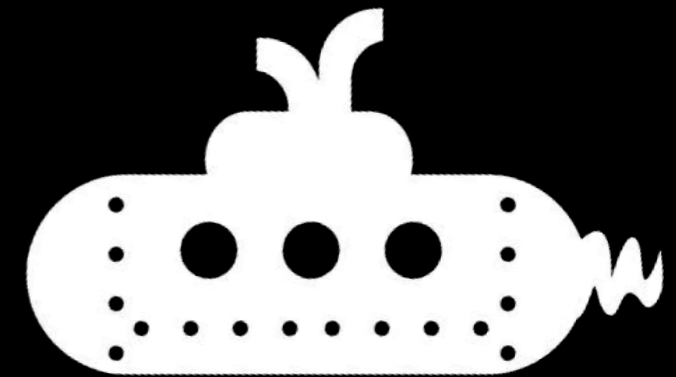
 Information

371

 Eliminate unnecessary not's ?   This is useless!

 Messages sent but not implemented ?  This is useless!

 Temporarily variables not read AND written ?  This is useless!



Nautilus



Quality Assistant

```
exportSettingAction
| targetSymbol currentValue defaultValue |

targetSymbol := self target isSymbol
ifTrue: [ self target ]
ifFalse: [ self target name asSymbol ].

currentValue := self targetSelector
ifNil: [ (Smalltalk at: targetSymbol)
        perform: self getSelector ]
ifNotNil: [:s |
          ((Smalltalk at: targetSymbol)
           perform: s) perform: self getSelector].

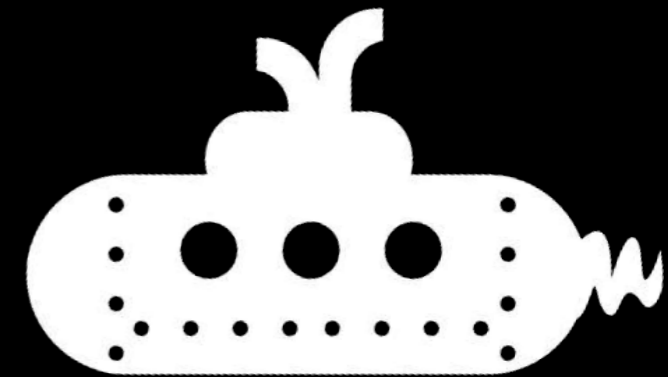
^ (self hasDefault not or: [
   (self default = currentValue) not ])
   ifTrue: [ self startupAction ]
   ifFalse: [ "We do nothing" nil ]
```

371

⚠ Eliminate unnecessary not's ? ✖ This is useless!

❗ Messages sent but not implemented ? ✖ This is useless!

ℹ Temporary variables not read AND written ? ✖ This is useless!




Nautilus



Quality Assistant







```
exportSettingAction
| targetSymbol currentValue defaultValue |
t
c
a
```

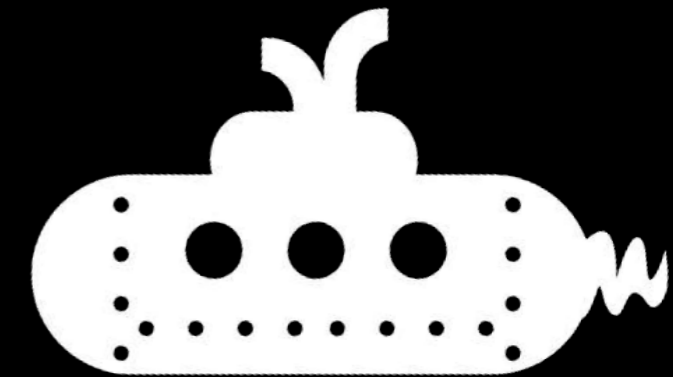
Messages sent but not implemented

 Checks for messages that are sent by a method, but no class in the system implements such a message. Further checks if messages sent to self or super exist in the hierarchy, since these can be statically typed. Reported methods will certainly cause a `doesNotUnderstand: message` when they are executed.

OK

371

-  Eliminate unnecessary not's ?  This is useless!
-  Messages sent but not implemented ?  This is useless!
-  Temporary variables not read AND written ?  This is useless!



Nautilus



Quality Assistant

```
exportSettingAction  
| targetSymbol currentValue defaultValue |
```

Apply the proposed changes?

Pretty print

```
on  
currentValue defaultValue |  
= self target isSymbol  
if target ]  
if target name asSymbol ].  
= self targetSelector  
lltalk at: targetSymbol) perform: self getSelector ]  
s | ((Smalltalk at: targetSymbol) perform: s) perform:  
ult not or: [ (self default = currentValue) not ]  
if startupAction ]  
nothing"
```

```
rentValue defaultValue |  
self target isSymbol  
target ]  
target name asSymbol ].  
if targetSelector  
lk at: targetSymbol) perform: self getSelector ]  
(Smalltalk at: targetSymbol) perform: s) perform  
not or: [ self default ~= currentValue ]  
artupAction ]  
hing"
```

Ok

Cancel

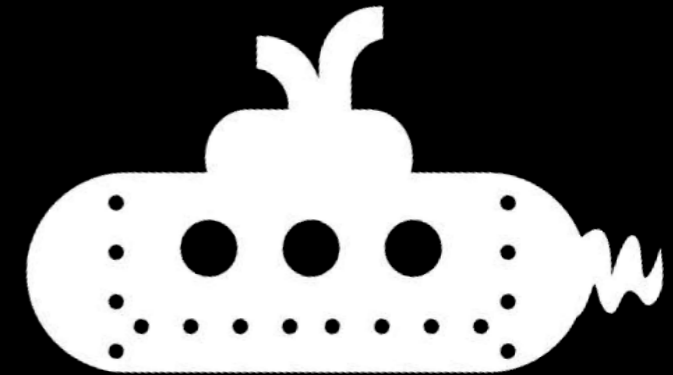
```
ifFalse: [ self startupAction ]  
ifFalse: [ "We do nothing" nil ]
```

371

⚠ Eliminate unnecessary not's ? ✖ This is useless!

⚠ Messages sent but not implemented ? ✖ This is useless!

ⓘ Temporary variables not read AND written ? ✖ This is useless!



Nautilus



Quality Assistant

```
exportSettingAction
| targetSymbol currentValue defaultValue |

targetSymbol := self target isSymbol
ifTrue: [ self target ]
ifFalse: [ self target name asSymbol ].

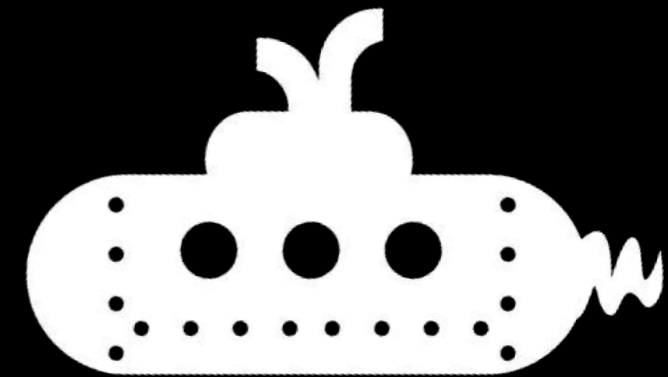
currentValue := self targetSelector
ifNil: [ (Smalltalk at: targetSymbol)
    perform: self getSelector ]
ifNotNil: [:s |
    ((Smalltalk at: targetSymbol)
    perform: s) perform: self getSelector]
```

On which level do you want to skip the rule?

SettingDeclaration>>#exportSettingAction
SettingDeclaration

Cancel

⚠ Eliminate unnecessary not's ? ✖ This is useless!
⚠ Messages sent but not implemented ? ✖ This is useless!
ⓘ Temporary variables not read AND written ? ✖ This is useless!



Nautilus



Quality Assistant

```
exportSettingAction
| targetSymbol currentValue defaultValue |

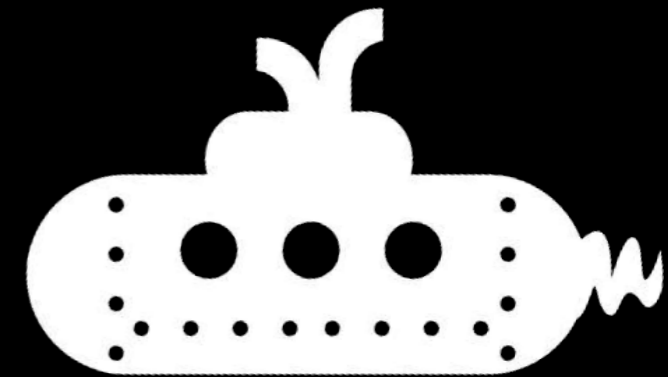
targetSymbol := self target isSymbol
ifTrue: [ self target ]
ifFalse: [ self target name asSymbol ].

currentValue := self targetSelector
ifNil: [ (Smalltalk at: targetSymbol)
    perform: self getSelector ]
ifNotNil: [:s |
    ((Smalltalk at: targetSymbol)
    perform: s) perform: self getSelector].

^ (self hasDefault not or: [
    (self default = currentValue) not ])
    ifTrue: [ self startupAction ]
    ifFalse: [ "We do nothing" nil ]
```

371

- ⚠ Eliminate unnecessary not's ? ✖ This is useless!
- ❗ Messages sent but not implemented ? ✖ This is useless!
- ℹ Temporary variables not read AND written ? ✖ This is useless!



Nautilus



Quality Assistant

```
exportSettingAction
| targetSymbol currentValue defaultValue |

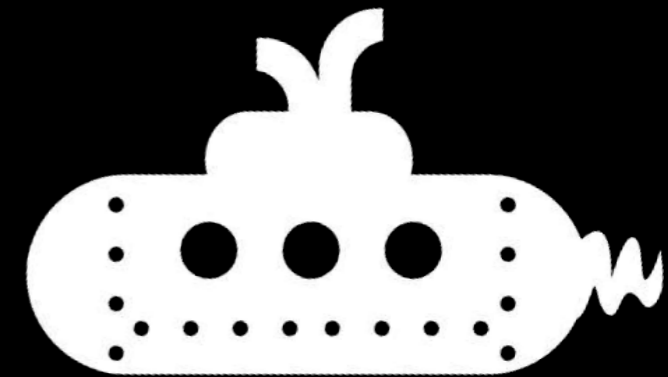
targetSymbol := self target isSymbol
ifTrue: [ self target ]
ifFalse: [ self target name asSymbol ].

currentValue := self targetSelector
ifNil: [ (Smalltalk at: targetSymbol)
        perform: self getSelector ]
ifNotNil: [:s |
          ((Smalltalk at: targetSymbol)
           perform: s) perform: self getSelector].

^ (self hasDefault not or: [
   (self default = currentValue) not ])
   ifTrue: [ self startupAction ]
   ifFalse: [ "We do nothing" nil ]
```

371

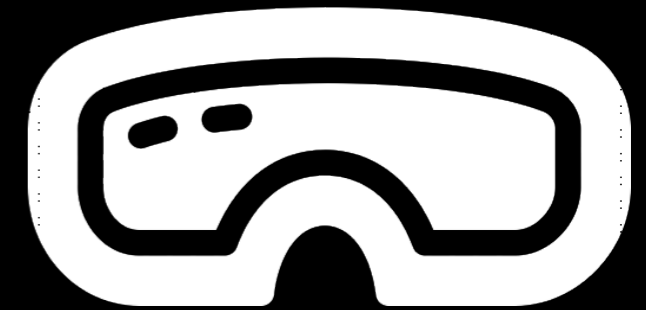
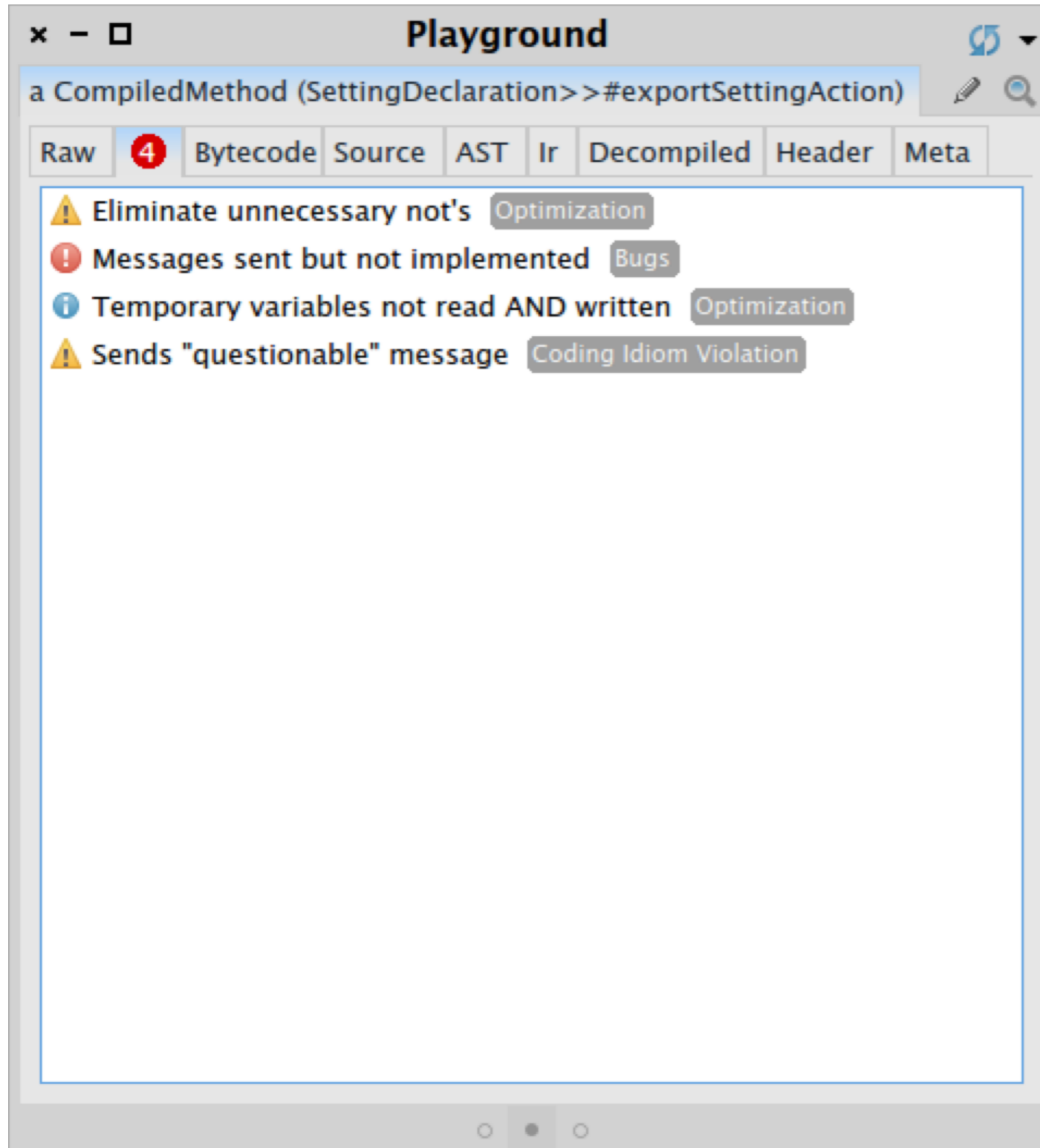
- ⚠ Eliminate unnecessary not's ? ✖ This is useless!
- ❗ Messages sent but not implemented ? ✖ This is useless!
- ℹ Temporary variables not read AND written ? ✖ This is useless!



Nautilus



Quality Assistant



Inspector



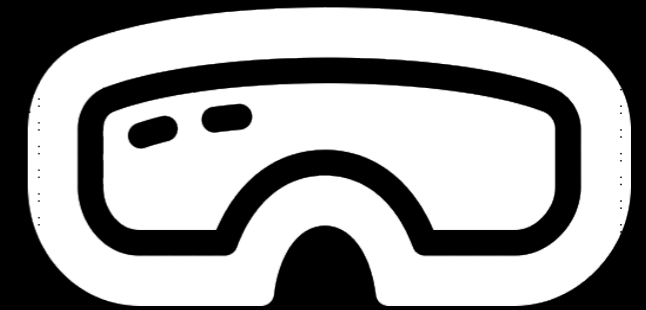
Quality Assistant

```
Playground
a ReCritic x
Source ... Description Raw Meta
exportSettingAction
  | targetSymbol currentValue defaultValue |

targetSymbol := self target isSymbol
  ifTrue: [ self target ]
  ifFalse: [ self target name asSymbol ].

currentValue := self targetSelector
  ifNil: [ (Smalltalk at: targetSymbol)
    perform: self getSelector ]
  ifNotNil: [:s |
    ((Smalltalk at: targetSymbol)
      perform: s) perform: self getSelector].

^ (self hasDefault not or: [
  (self default = currentValue) not ])
  ifTrue: [ self startupAction ]
  ifFalse: [ "We do nothing" nil ]
```



Inspector



Quality Assistant

SettingDeclaration>>#expc...

Search



x

QA Critics 4/4 →

- ⚠ Eliminate unnecessary not's
- ℹ Temporary variables not read AND w...
- ⚠ Sends "questionable" message
- ❗ Messages sent but not implemented ✓ →

Senders 1/1

SettingBrowser>>#exportSettings

Implementors 2/2

PragmaSetting>>#exportSettingAction

SettingDeclaration>>#exportSetting...

Messages sent but not implemented

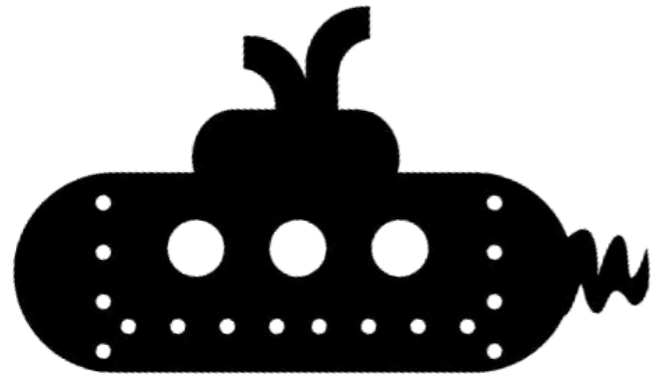
Checks for messages that are sent by a method, but no class in the system implements such a message. Further checks if messages sent to self or super exist in the hierarchy, since these can be statically typed. Reported methods will certainly cause a `doesNotUnderstand: message` when they are executed.



Spotter



Quality Assistant



Nautilus



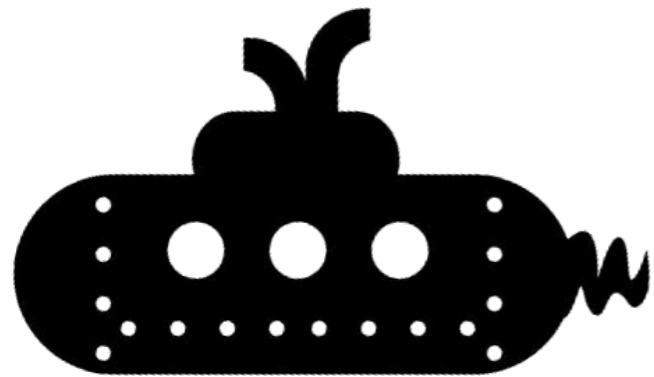
Inspector



Spotter



Quality Assistant



Nautilus

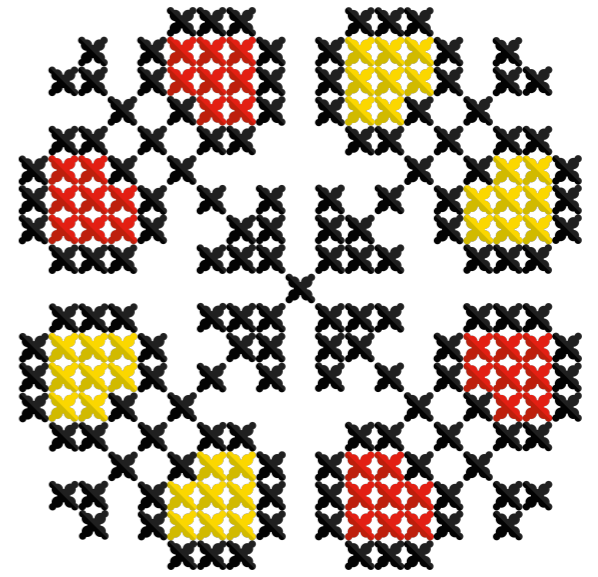


Inspector

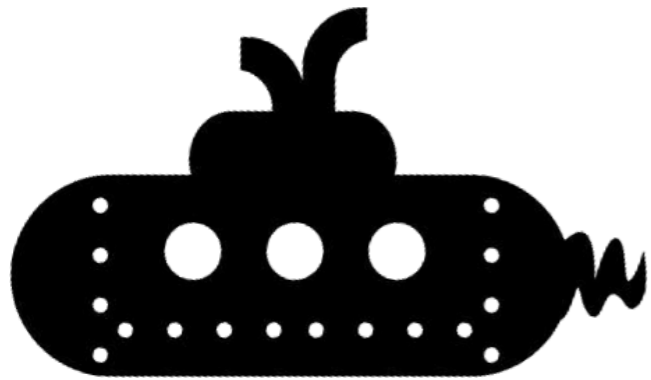


Spotter

Renraku



Quality Assistant



Nautilus

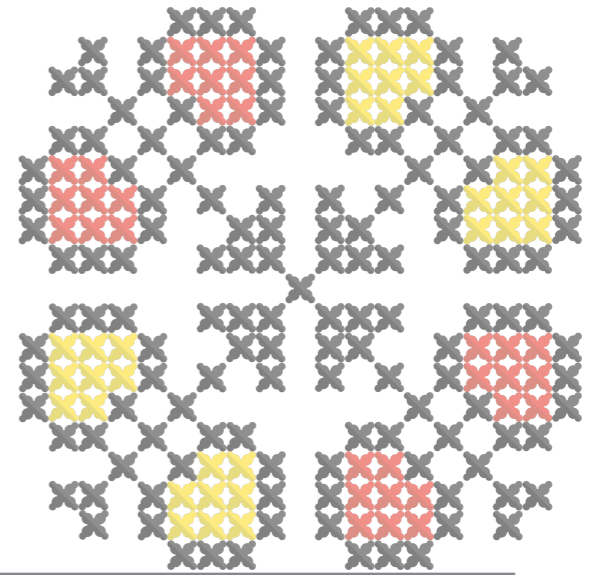


Inspector



Spotter

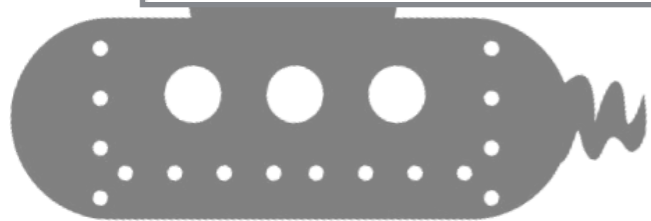
Renraku



```
someWeirdStuff  
  
^ true = true | flase.  
  ^ true = true | flase.  
  ^ true = true | flase.
```

! Declare: Temporary Variable, Instance Variable
Suggestions: false

? Unnecessary "= true"



Nautilus

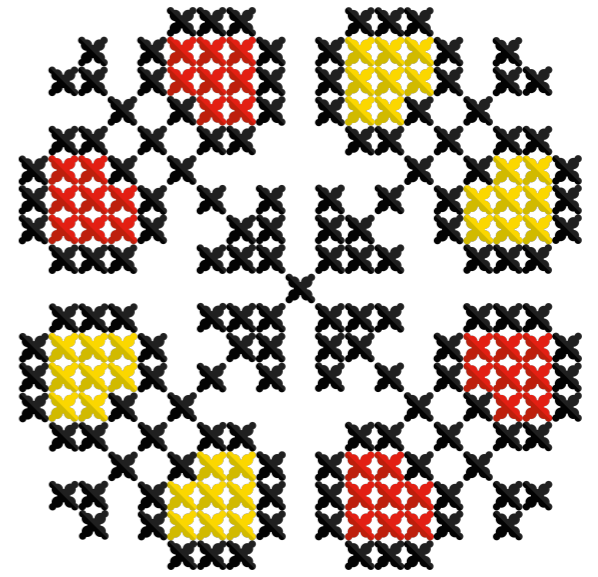


Inspector

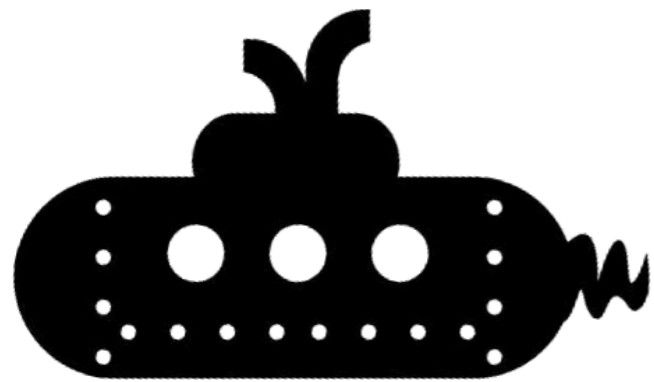


Spotter

Renraku



Quality Assistant



Nautilus

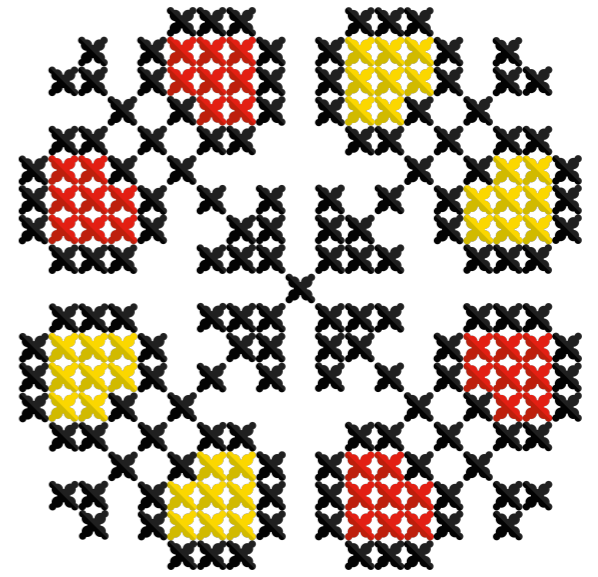


Inspector



Spotter

Renraku



Quality Assistant



Nautilus+



Inspector



Spotter



Debugger



Diff



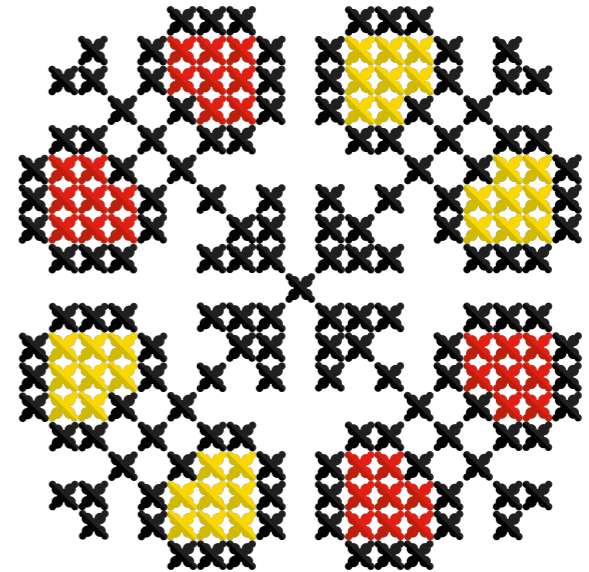
CI Server



CriticBrowser

<https://github.com/Uko/QualityAssistant>

Renraku



Quality Assistant



Nautilus+



Inspector



Spotter



Debugger



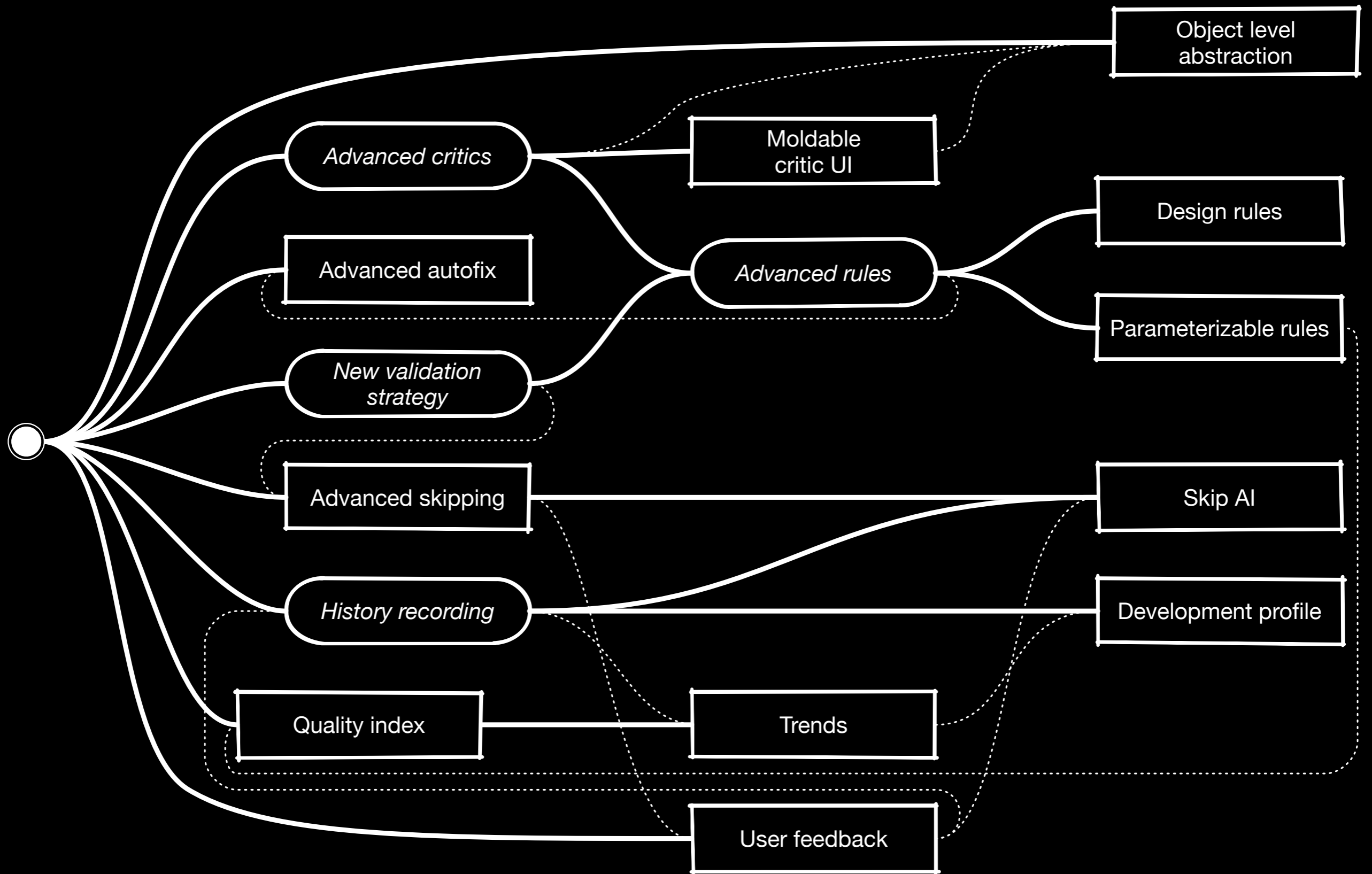
Diff



CI Server



CriticBrowser



User feedback

```
prepareRenderElement: edgeElement stateTracker: stateTracker  
| builder vector1 vector2 point |  
builder := self lineBuilder: stateTracker.  
vector1 := edgeElement from position.  
vector2 := edgeElement to position.
```

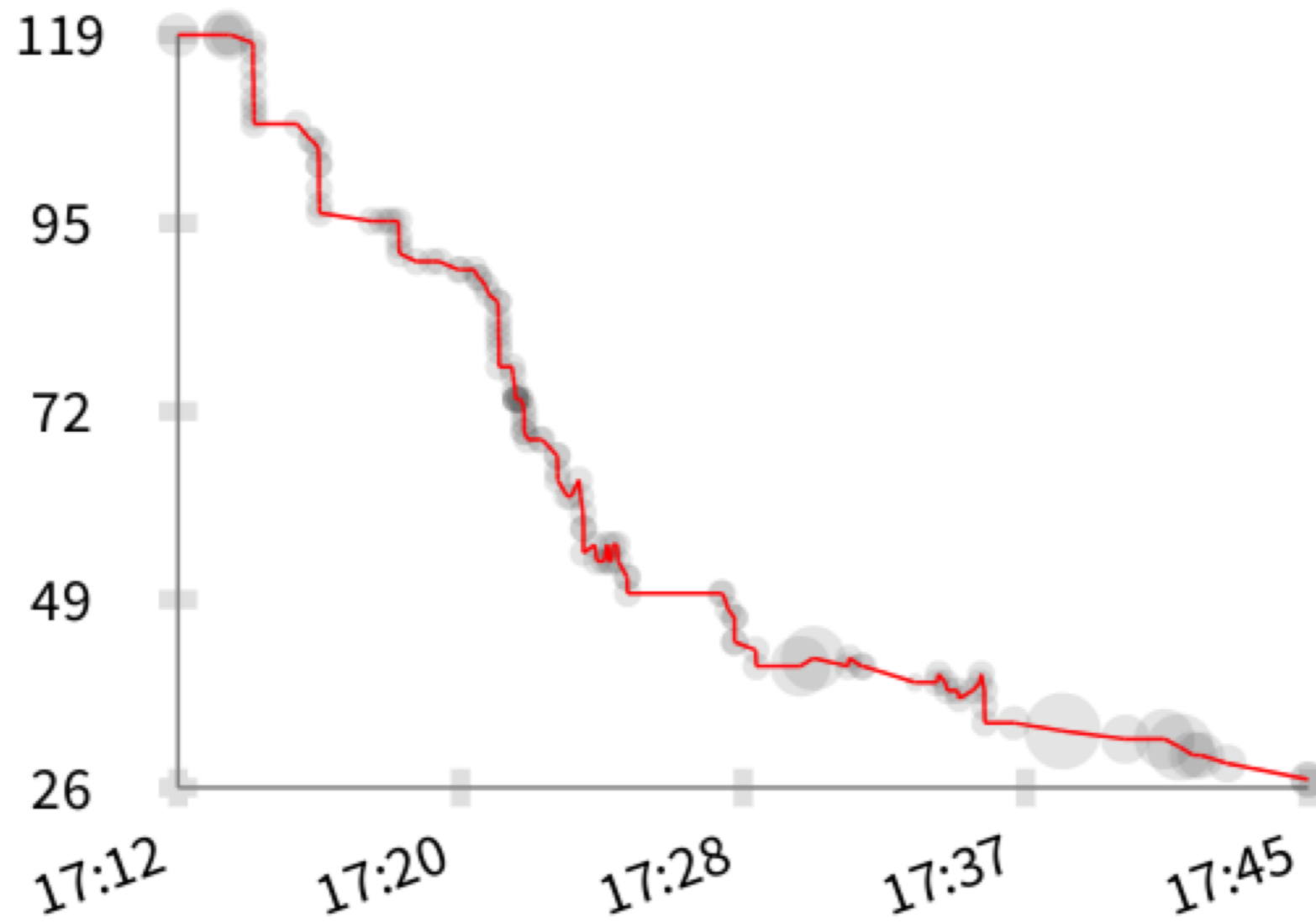
361

- ⚠ Eliminate unnecessary not's ? ✖ This is useless!
- ❗ Messages sent but not implemented ? ✖ This is useless!
- ℹ Temporary variables not read AND written ? ✖ This is useless!

<http://renraku.inf.usi.ch>

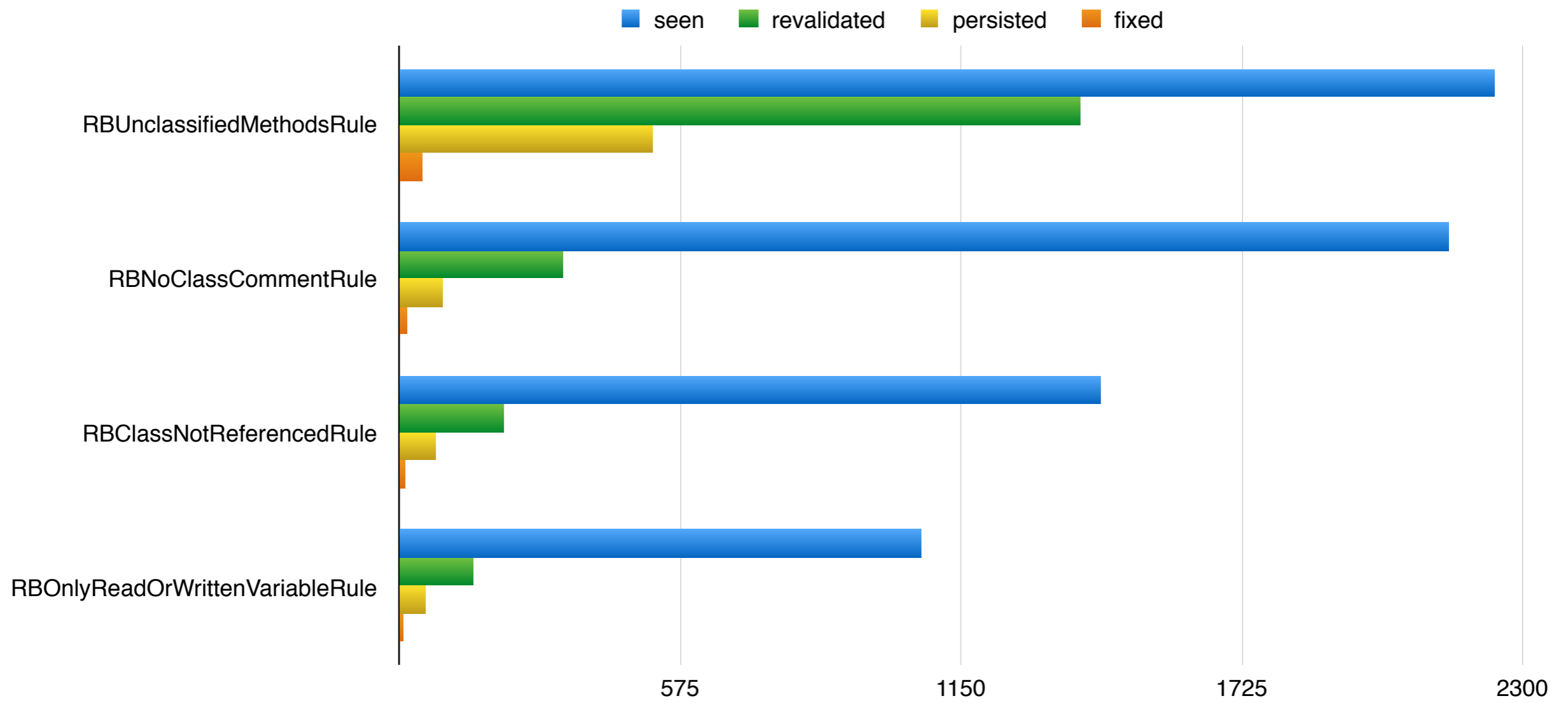
Quality index

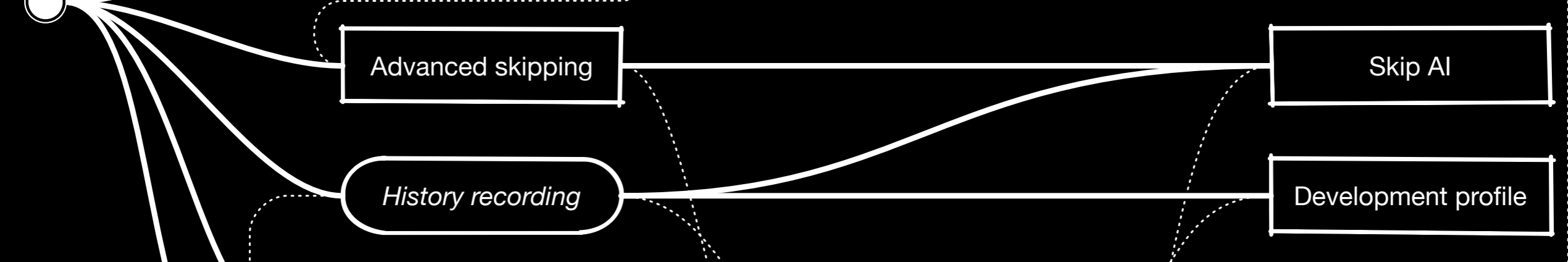
Trends



History recording

Development profile



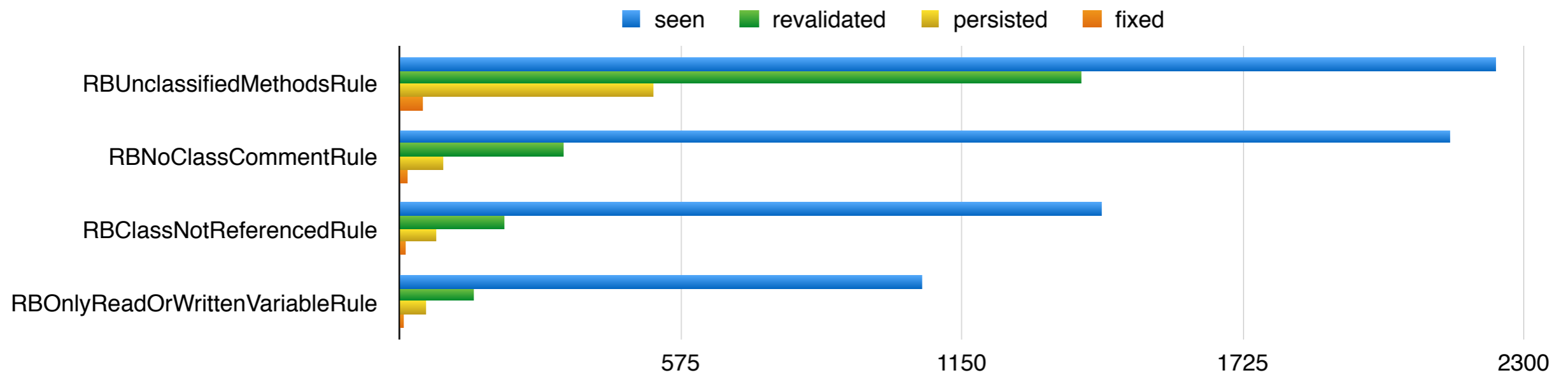


```

    currentValue := self targetSelector
    ifNil: [ (Smalltalk at: targetSymbol)
            perform: self getSelector ]
    ifNotNil: [ ... ]
  
```

371

- ⚠ Eliminate unnecessary not's ? ✖ This is useless!
- ⚠ Messages sent but not implemented ? ✖ This is useless!
- ⚠ Sends "questionable" message ? ✖ This is useless!



Advanced autofix

Ad

```
if target ]  
if target name asSymbol ].  
self targetSelector  
Smalltalk at: targetSymbol) perform: self getSelector ]  
s | ((Smalltalk at: targetSymbol) perform: s) perform:  
ult not or: [ (self default = currentValue) not ]  
if startupAction ]  
nothing"
```

```
target ]  
target name asSymbol ].  
if targetSelector  
Smalltalk at: targetSymbol) perform: self getSelector ]  
((Smalltalk at: targetSymbol) perform: s) perform  
not or: [ self default ~= currentValue ]  
startupAction ]  
hing"
```

Ok

Cancel

```
ifFalse: [ "We do nothing" nil ]
```

371

⚠ Eliminate unnecessary not's ?  ✖ This is useless!

⚠ Messages sent but not implemented ? ✖ This is useless!

ⓘ Temporary variables not read AND written ? ✖ This is useless!



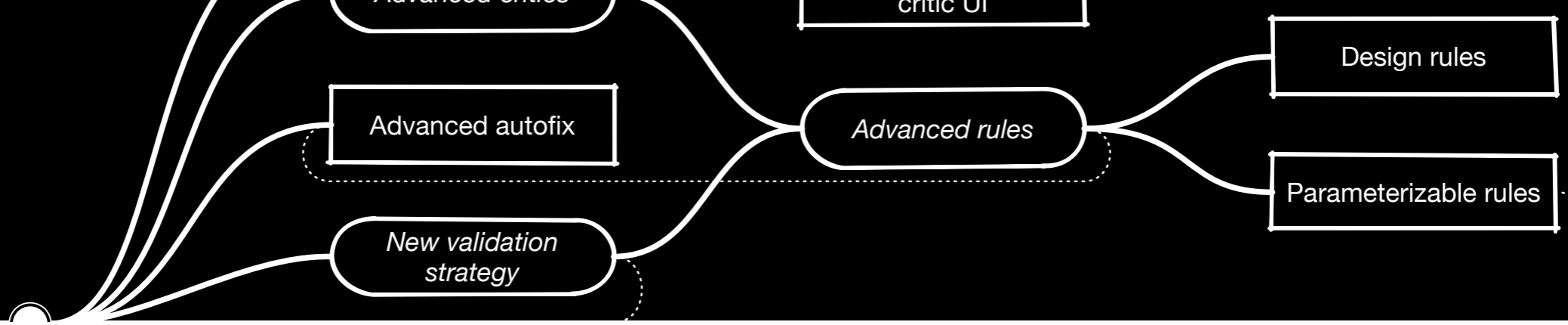
Advanced critics

Moldable critic UI

The screenshot displays the 'Inspector on a RPackage(Kernel)' interface. The top bar shows the package name 'a RPackage(Kernel)' and the selected object 'an Integer class (Integer)'. Below the top bar, there are three main panels:

- Box Hierarchy:** A tree view showing the structure of the R package, with various boxes representing different components.
- Viz Me...:** A visualization of the selected object, showing a large gray square.
- Disorder Radar:** A radar chart with three axes: 'Number of temporaries' (top), 'Number of parameters' (right), and 'Number of statements' (bottom). The chart shows a blue shaded area representing the distribution of these metrics. The axes are labeled with values 0.0, 0.25, 0.5, 0.75, and 1.0.

The Disorder Radar chart shows a blue shaded area representing the distribution of metrics. The axes are labeled with values 0.0, 0.25, 0.5, 0.75, and 1.0. The chart shows a blue shaded area representing the distribution of metrics.



ArgoUML Report

Index

- ▼ ArgoUML design pro
 - Non-private attri
 - Deprecated class
 - Deprecated meth
- ▼ ArgoUML naming pr
 - Classes with too
- ▼ To watch for
 - Facade watcher

Map of packages containing Facade classes

Facade watcher

Package map

```

      graph TD
        kernel --- taskmgmt
        kernel --- sequence2 --- diagram
        kernel --- util --- logging
        kernel --- util --- osdep
        kernel --- i18n
        kernel --- persistence
        kernel --- notation --- providers --- java
        kernel --- notation --- providers --- uml
        kernel --- notation --- ui
        kernel --- moduleloader
        kernel --- ocl
        kernel --- swingext
        kernel --- cognitive --- critics --- ui
        kernel --- cognitive --- checklist --- ui
        kernel --- cognitive --- ui
        kernel --- configuration
        kernel --- profile --- init
        kernel --- profile --- internal --- ocl --- uml14
        kernel --- profile --- internal --- ui
        kernel --- model --- euml
        kernel --- model --- mdr
        kernel --- gefext
        kernel --- application --- apl
        kernel --- application --- events
        kernel --- application --- security
        kernel --- application --- helpers
      
```

Facade class

```

class FacadeEUMLImpl implements Facade
{
    /**
     * The model implementation.
     */
    private EUMLModelImplementation
    modelImpl;

    /**
     * Constructor.
     *
     * @param implementation
     *       The ModelImplementation.
     */
    public
    FacadeEUMLImpl(EUMLModelImplementatio
    n implementation) {
        modelImpl = implementation;
    }
}
      
```

Object level
abstraction

Moldable

The screenshot shows a Playground IDE window titled "Playground". The interface is split into two main panes. The left pane displays a list of warnings for the current code, with "Messages sent but not implemented" selected. The right pane shows the source code for a class named "ReCritic".

Warnings:

- Eliminate unnecessary not's (Optimization)
- Messages sent but not implemented (Bugs)
- Temporary variables not read AND written (Optimization)
- Sends "questionable" message (Coding Idiom Violation)

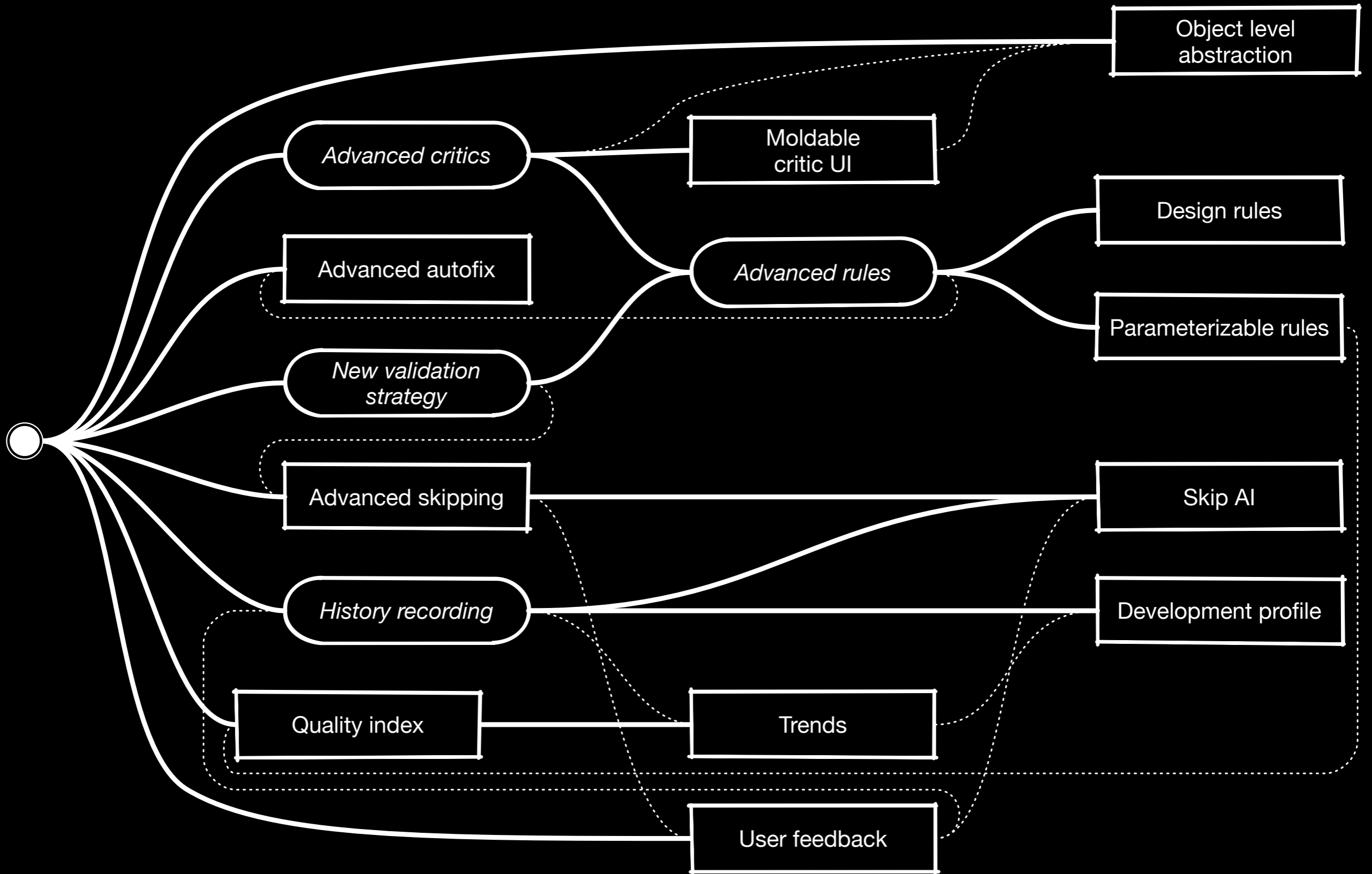
Source Code:

```
exportSettingAction
| targetSymbol currentValue defaultValue |

targetSymbol := self target isSymbol
ifTrue: [ self target ]
ifFalse: [ self target name asSymbol ].

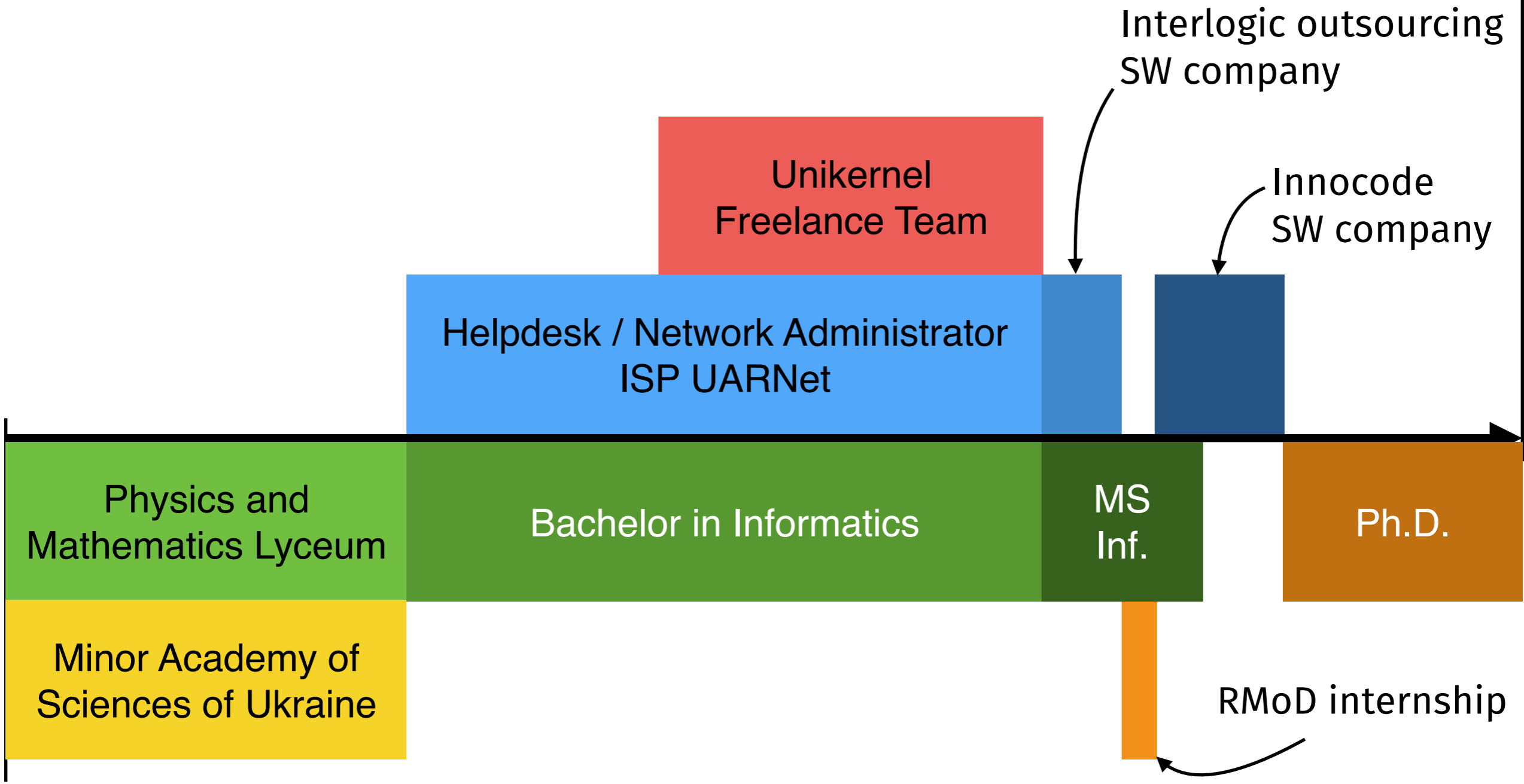
currentValue := self targetSelector
ifNil: [ (Smalltalk at: targetSymbol)
perform: self getSelector ]
ifNotNil: [:s |
((Smalltalk at: targetSymbol)
perform: s) perform: self getSelector].

^ (self hasDefault not or: [
(self default = currentValue) not ])
ifTrue: [ self startupAction ]
ifFalse: [ "We do nothing" nil ]
```



Now. YT 24yo 

<http://yuriy.tymch.uk>



2006. YT 15yo 