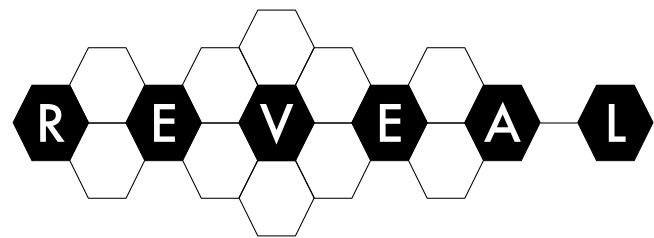
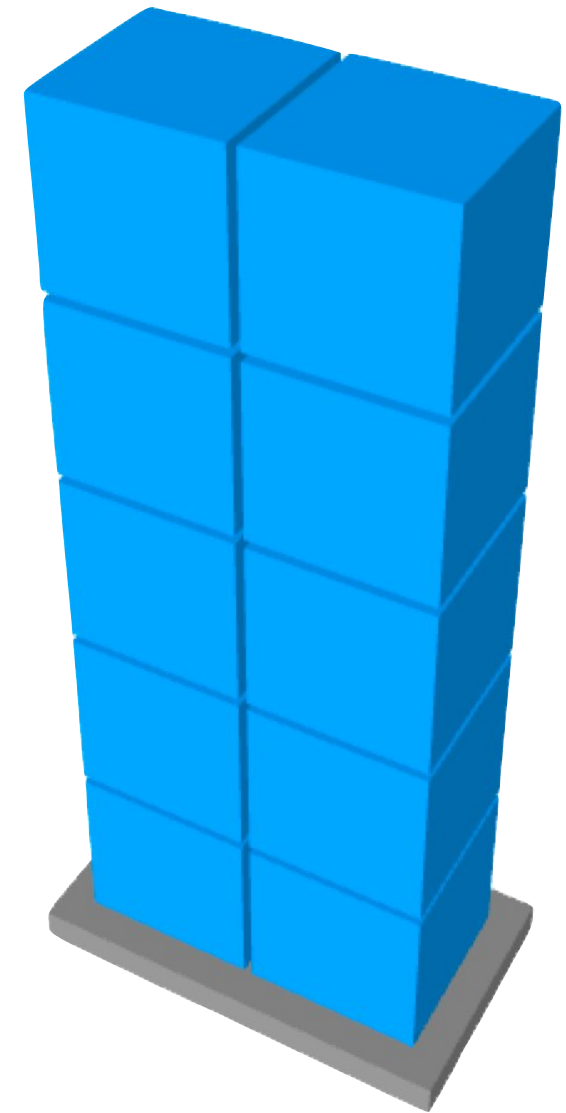
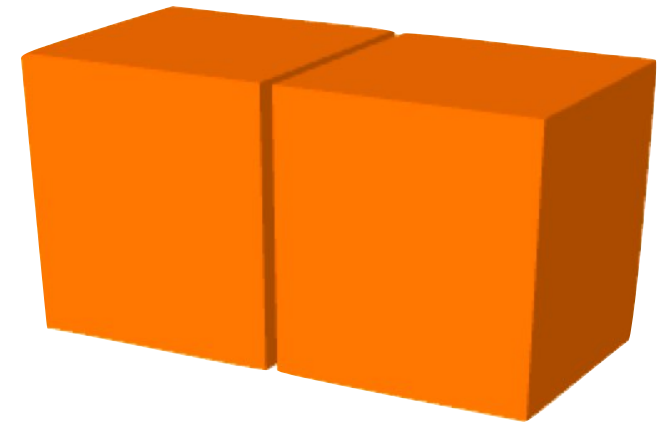


ViDI

Visual Design Inspector

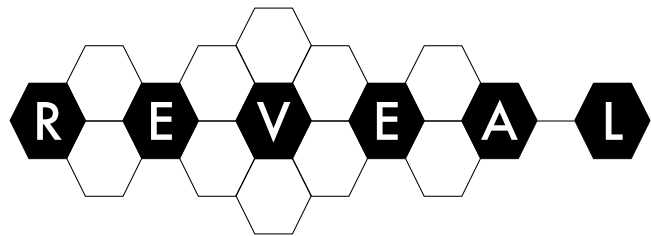


Yuriy Tymchuk
Michele Lanza

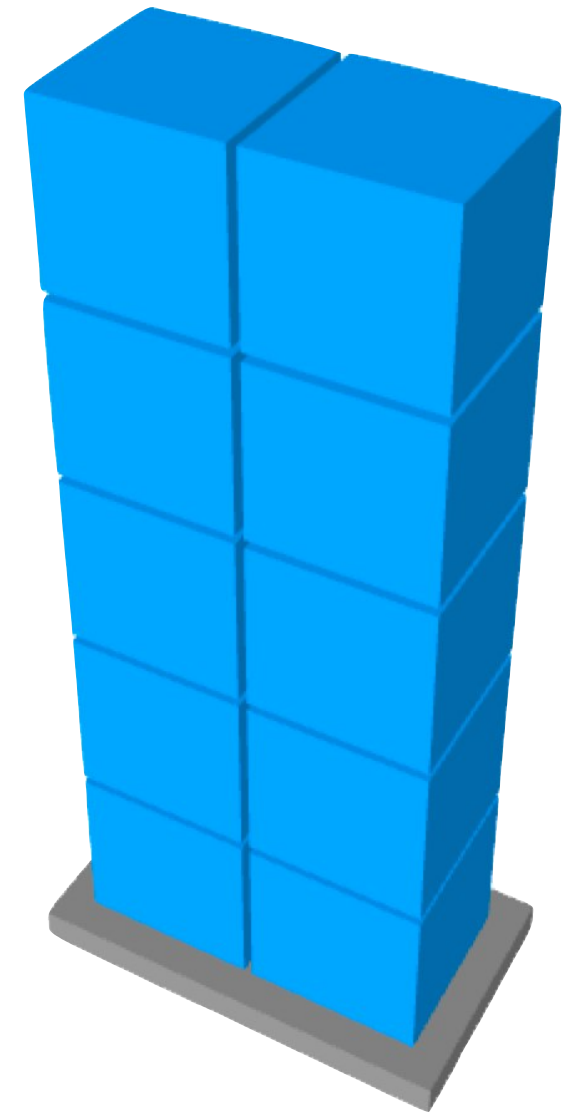
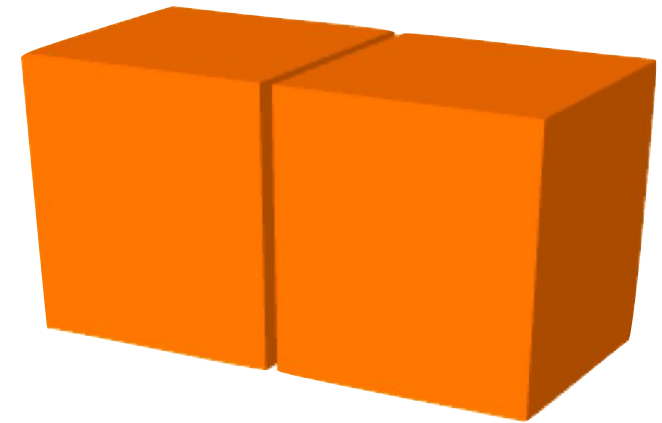


ViDI

Visual ~~Design~~ Inspector
Defect

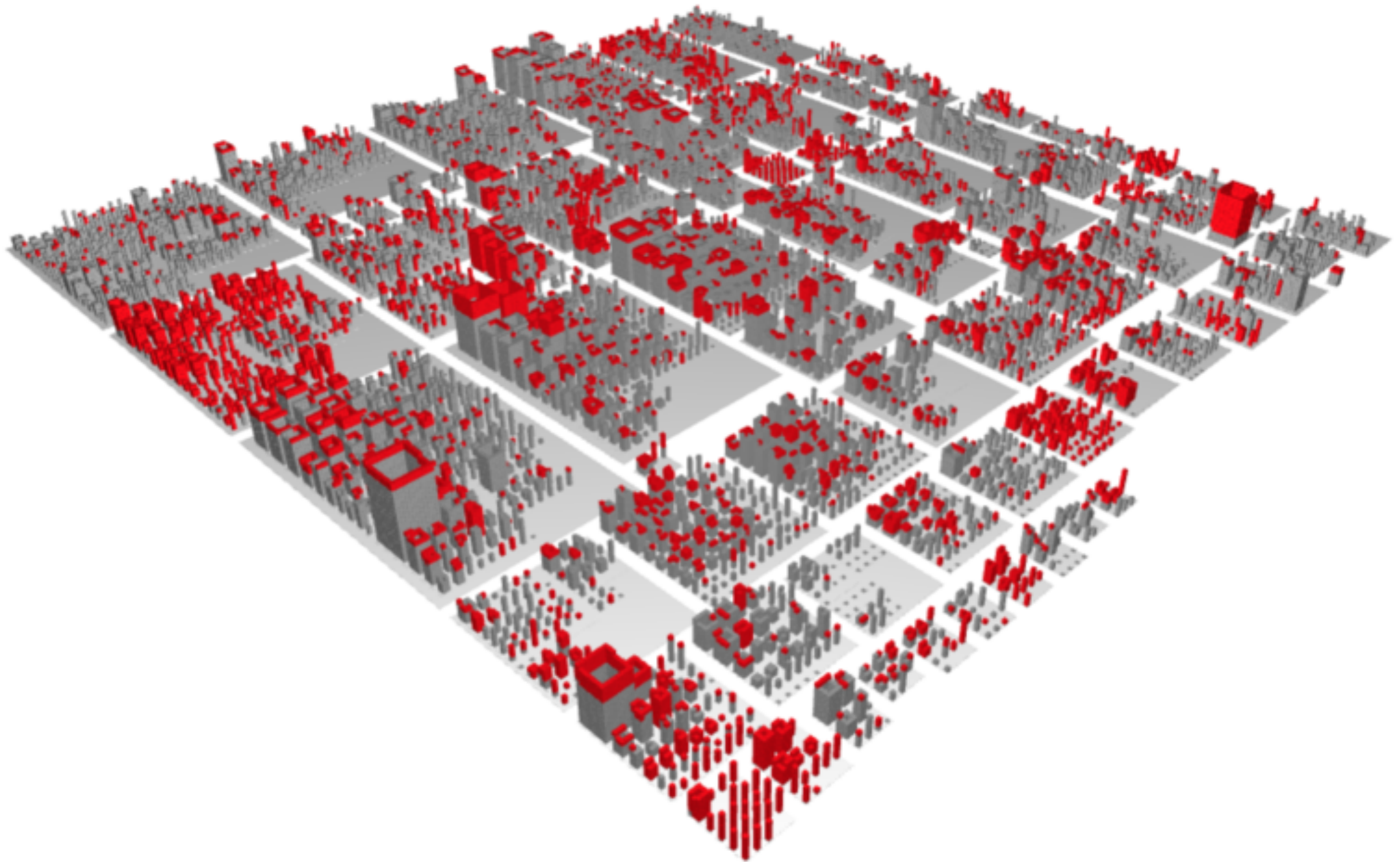


Yuriy Tymchuk
Michele Lanza

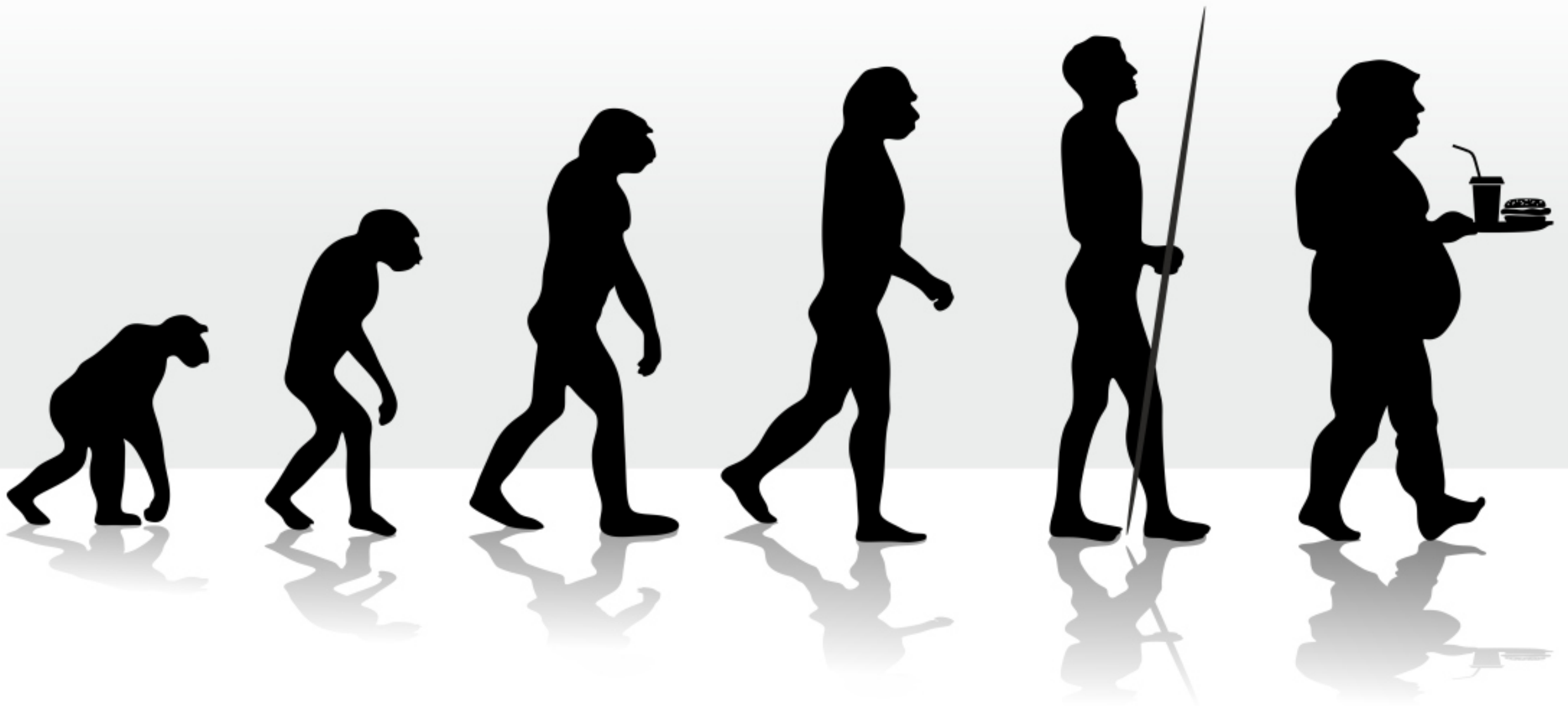


Not so long ago
in the cyber space ...
promoted by Doru

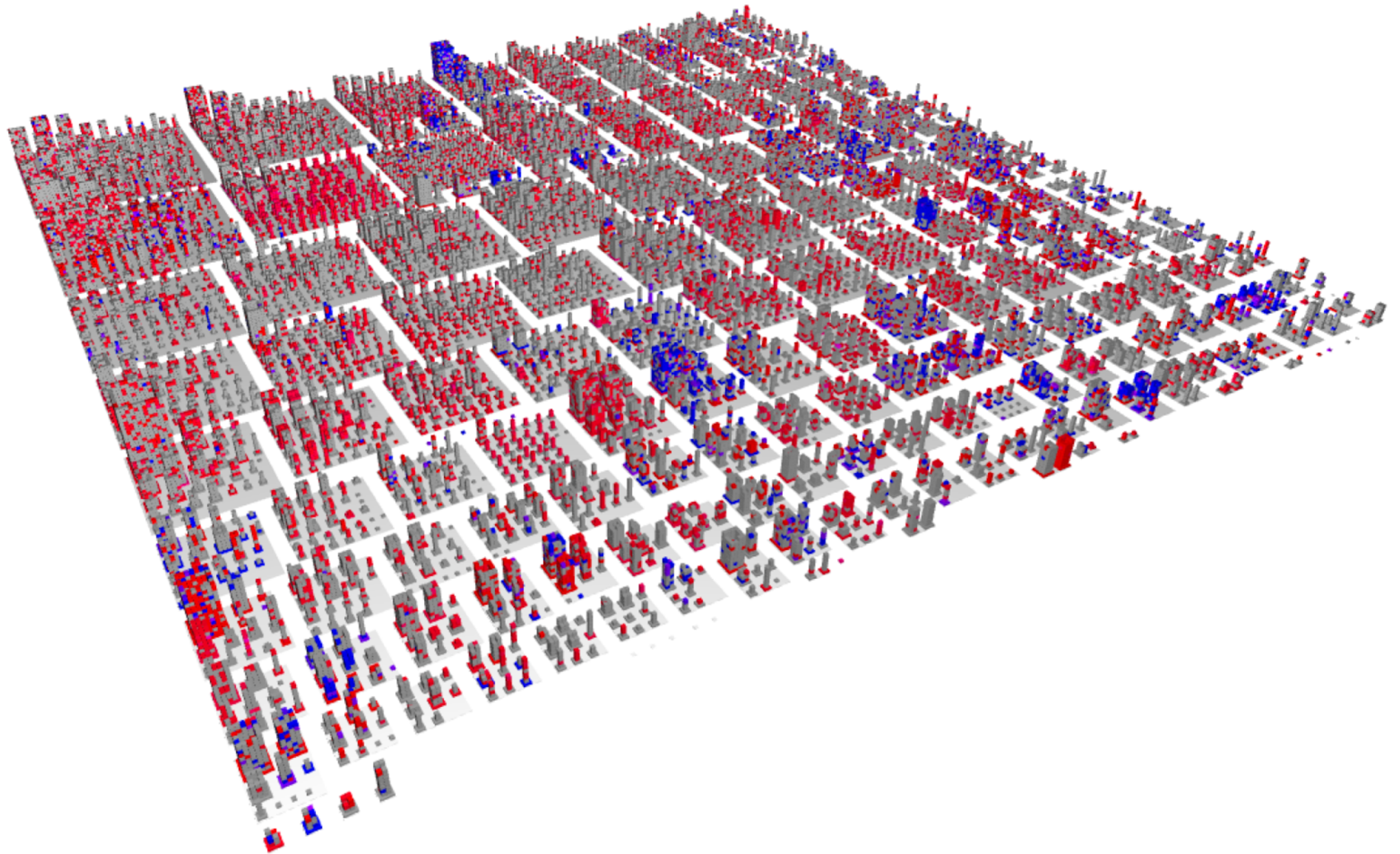
Pharo 3 Changes



Changes...



Pharo 2-3 Critiques





Unknown 06 June, 2014 17:16

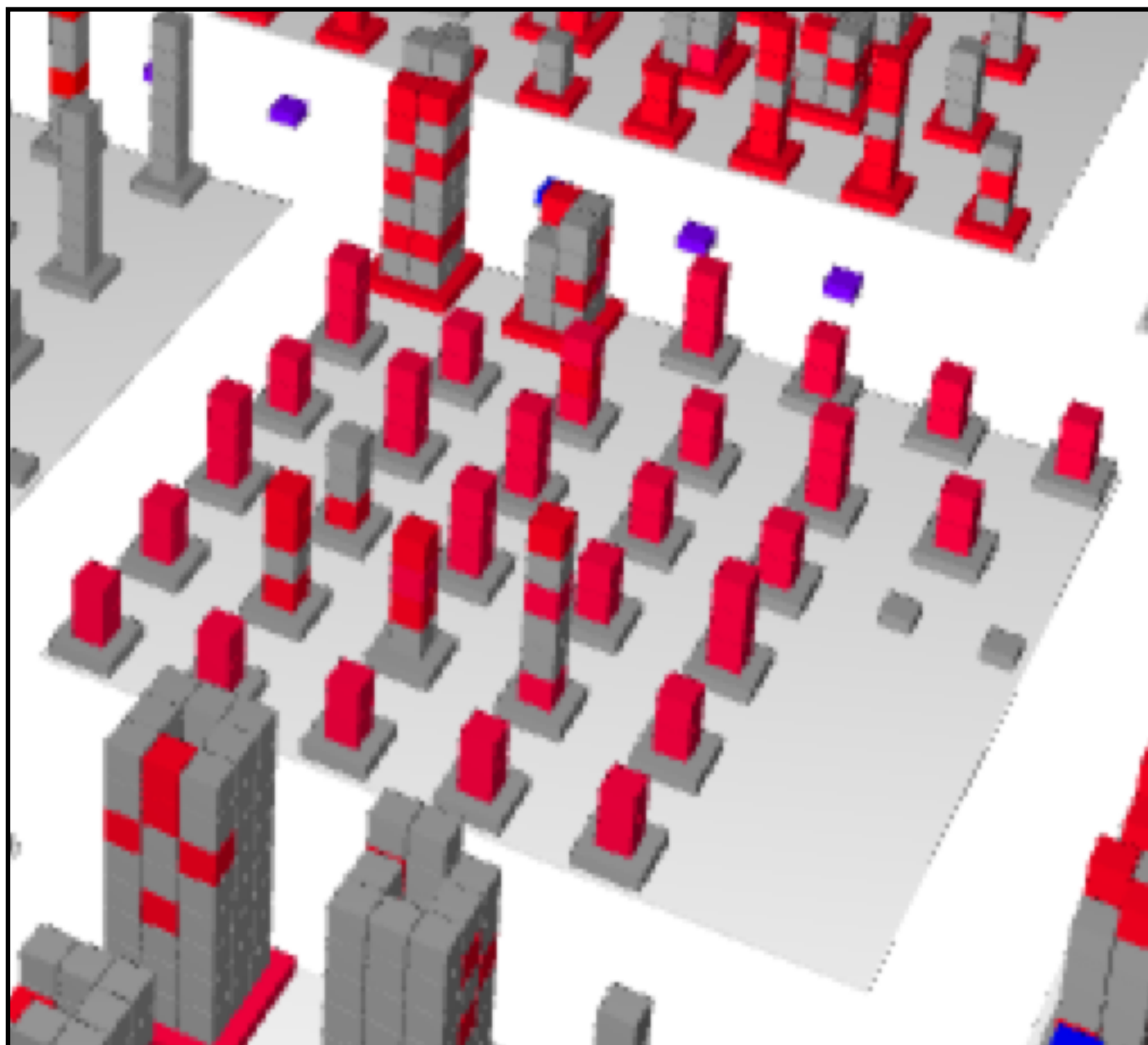
SmartSuggestions was introduced in Pharo3... so of course it went from "no code critique" to "lots of problems".

All the blue comes from those (not many) cases that someone actually did sit down and painstakingly fix the issues... last but not least: there are *lots* of false positives and things that are not things that people ever will fix.

We need to distinguish code critique rules that are to be fixed from those that, statistically, may hint problematic design when seen a lot, but that are not clear "errors" that would be fixable.

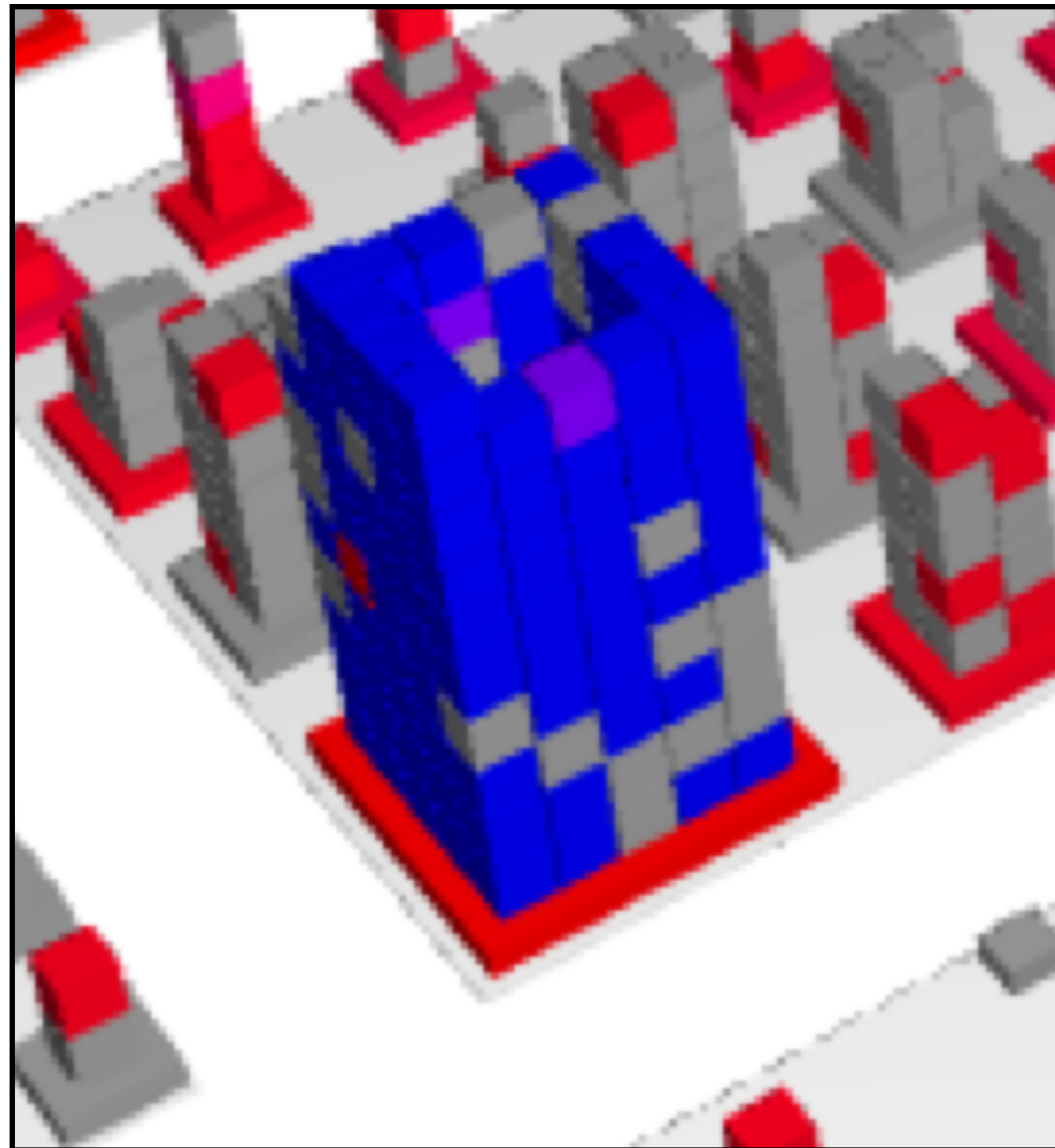
And for tests: a lot of "smells" are actually not smells for tests. People therefore never run the code critique for tests... e.g. "lots of methods". Is having lots of test methods in a test really bad?

I can only advise everyone who tries to do research with code critique to actually use it a bit... it's more an accidental set of random things, not something that has intrinsic value at this state.



Unknown 06 June, 2014 17:16

SmartSuggestions was introduced in Pharo3... so of course it went from "no code critique" to "lots of problems".



All the blue comes from those (not many) cases that someone actually did sit down and painstakingly fixed the issues... last but not least: there are *lots* of false positives and things that are not things that people ever will fix.



And for tests: a lot of "smells" are actually not smells for tests. People therefore never run the code critique for tests... e.g. "lots of methods". Is having lots of test methods in a test really bad?

<ul style="list-style-type: none"> ▼ Selected Rules (To sort: 40, ToDo: 0, Wrong: 21) <ul style="list-style-type: none"> ▼ Bugs (To sort: 9, ToDo: 0, Wrong: 0) <ul style="list-style-type: none"> Debugging code left in methods (To sort: 0, ToDo: 0, Wrong: 0) Messages sent but not implemented (To sort: 8, ToDo: 0, Wrong: 0) Method source contains linefeeds (To sort: 0, ToDo: 0, Wrong: 0) No direct access of methodDict (To sort: 1, ToDo: 0, Wrong: 0) Overrides a "special" message (To sort: 0, ToDo: 0, Wrong: 0) References an undeclared variable (To sort: 0, ToDo: 0, Wrong: 0) Sends super new initialize (To sort: 0, ToDo: 0, Wrong: 0) Sends unknown message to global (To sort: 0, ToDo: 0, Wrong: 0) Subclass responsibility not defined (To sort: 0, ToDo: 0, Wrong: 0) Uses True/False instead of true/false (To sort: 0, ToDo: 0, Wrong: 0) Variable used, but not defined anywhere (To sort: 0, ToDo: 0, Wrong: 0) ▶ Optimization (To sort: 3, ToDo: 0, Wrong: 6) ▶ Potential Bugs (To sort: 0, ToDo: 0, Wrong: 1) ▶ Design Flaws (To sort: 10, ToDo: 0, Wrong: 8) ▼ Unclassified rules (To sort: 0, ToDo: 0, Wrong: 0) <ul style="list-style-type: none"> Check for same statements in ifTrue:iffalse: blocks (To sort: 0, ToDo: 0, Wrong: 0) Code Rewriting (To sort: 0, ToDo: 0, Wrong: 0) Explicit requirement methods (To sort: 0, ToDo: 0, Wrong: 0) Repeated method in the superclasses (To sort: 0, ToDo: 0, Wrong: 0) Repeated methods in the trait composition (To sort: 0, ToDo: 0, Wrong: 0) ▶ Coding Idiom Violation (To sort: 14, ToDo: 0, Wrong: 6) ▶ Style (To sort: 4, ToDo: 0, Wrong: 0) 	<ul style="list-style-type: none"> ❗ CriticBrowser>>#methodAdded: (Manifest-CriticBrowser) ❗ SelectBrowser>>#items: (Manifest-CriticBrowser) ❗ SelectBrowser>>#setAddAllButtonModel (Manifest-CriticBrowser) ❗ SelectBrowser>>#setAddButtonModel (Manifest-CriticBrowser) ❗ SelectBrowser>>#setRemoveAllButtonModel (Manifest-CriticBrowser) ❗ SelectBrowser>>#setRemoveButtonModel (Manifest-CriticBrowser) ❗ TApplyingOnClassSide>>#criticTheNonMetaclassClass (Traits-Kernel-Traits) ❗ TClassDescription>>#criticNameOn: (Traits-Kernel-Traits)
---	--

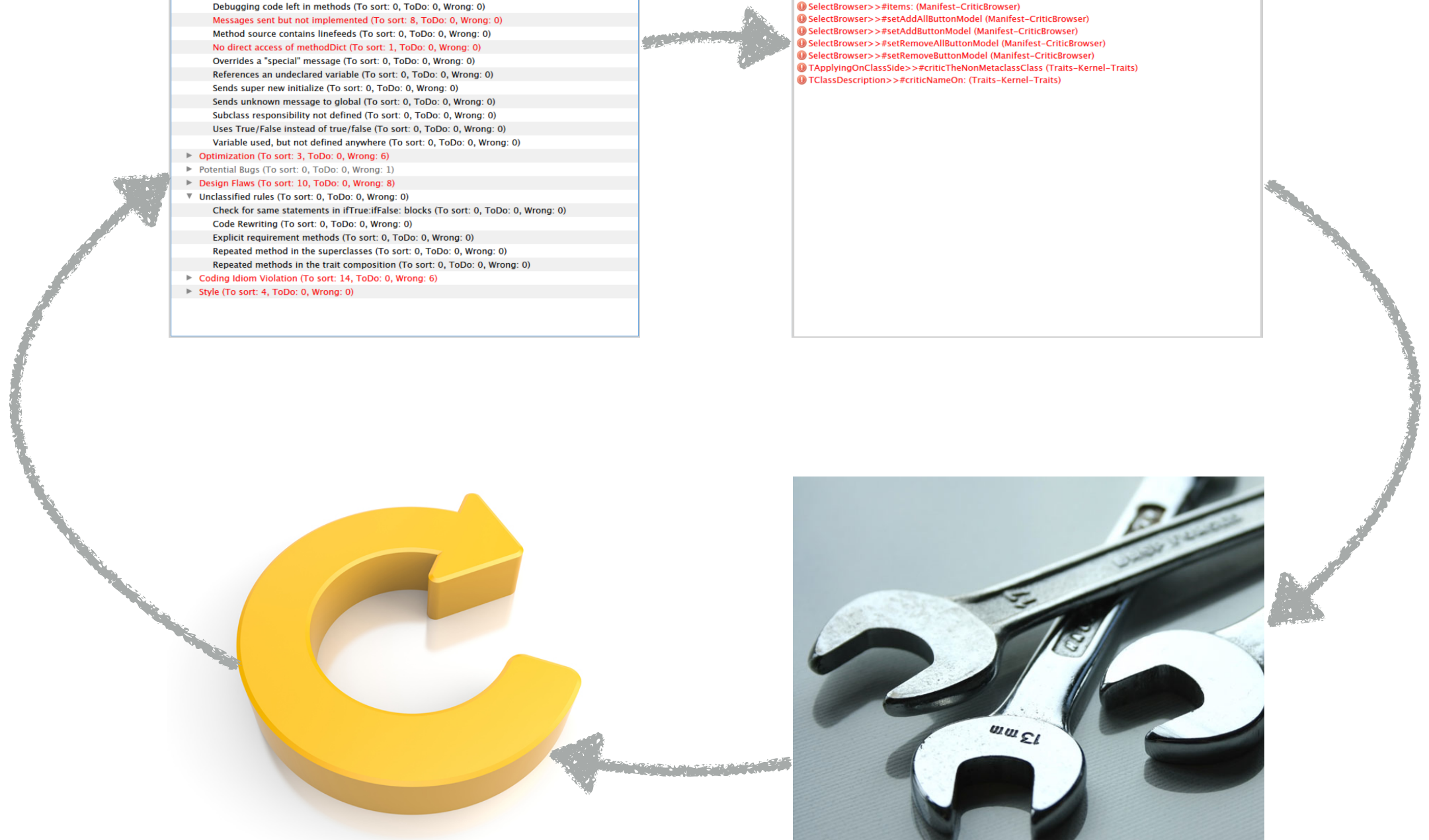
I can only advise everyone who tries to do research with code critique to actually use it a bit... it's more an accidental set of random things, not something that has intrinsic value at this state.

CC Workflow

Selected Rules (To sort: 40, ToDo: 0, Wrong: 21)

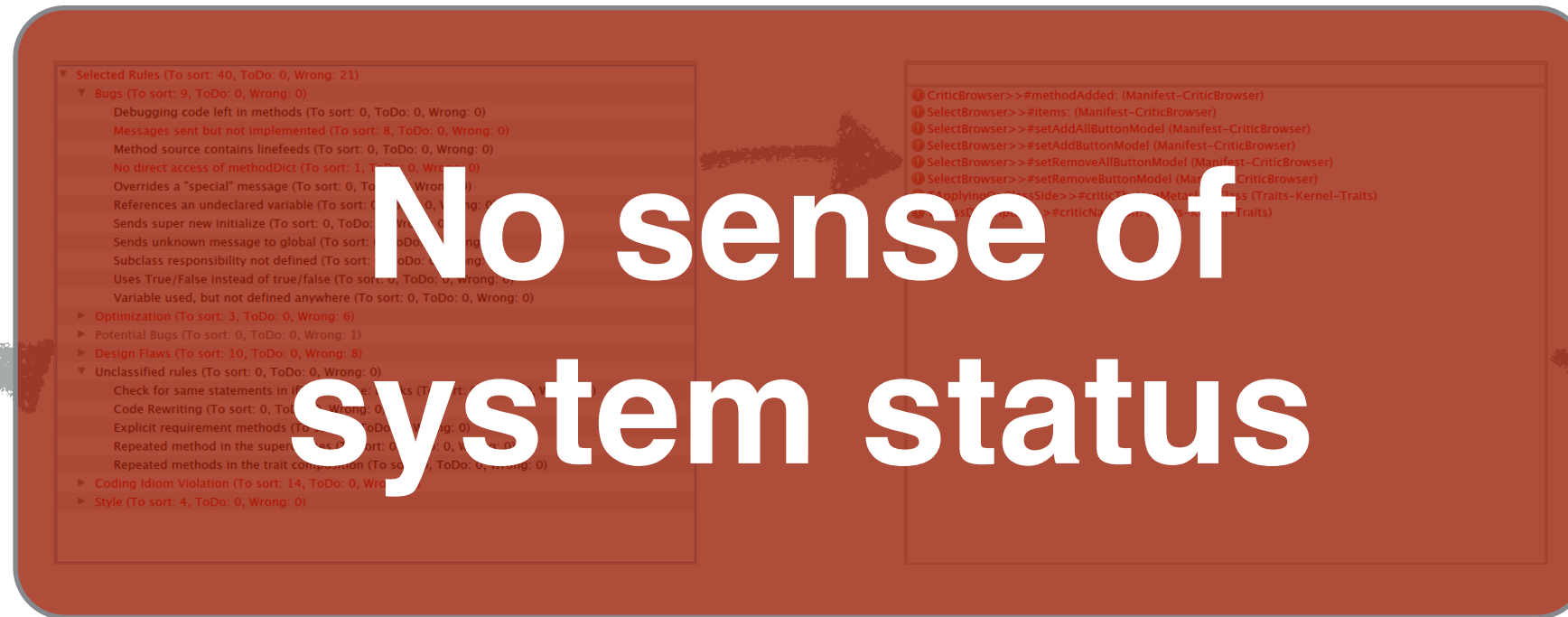
- ▼ Bugs (To sort: 9, ToDo: 0, Wrong: 0)
 - Debugging code left in methods (To sort: 0, ToDo: 0, Wrong: 0)
 - Messages sent but not implemented (To sort: 8, ToDo: 0, Wrong: 0)
 - Method source contains linefeeds (To sort: 0, ToDo: 0, Wrong: 0)
 - No direct access of methodDict (To sort: 1, ToDo: 0, Wrong: 0)
 - Overrides a "special" message (To sort: 0, ToDo: 0, Wrong: 0)
 - References an undeclared variable (To sort: 0, ToDo: 0, Wrong: 0)
 - Sends super new initialize (To sort: 0, ToDo: 0, Wrong: 0)
 - Sends unknown message to global (To sort: 0, ToDo: 0, Wrong: 0)
 - Subclass responsibility not defined (To sort: 0, ToDo: 0, Wrong: 0)
 - Uses True/False instead of true/false (To sort: 0, ToDo: 0, Wrong: 0)
 - Variable used, but not defined anywhere (To sort: 0, ToDo: 0, Wrong: 0)
- ▶ Optimization (To sort: 3, ToDo: 0, Wrong: 6)
- ▶ Potential Bugs (To sort: 0, ToDo: 0, Wrong: 1)
- ▶ Design Flaws (To sort: 10, ToDo: 0, Wrong: 8)
- ▼ Unclassified rules (To sort: 0, ToDo: 0, Wrong: 0)
 - Check for same statements in ifTrue:ifFalse: blocks (To sort: 0, ToDo: 0, Wrong: 0)
 - Code Rewriting (To sort: 0, ToDo: 0, Wrong: 0)
 - Explicit requirement methods (To sort: 0, ToDo: 0, Wrong: 0)
 - Repeated method in the superclasses (To sort: 0, ToDo: 0, Wrong: 0)
 - Repeated methods in the trait composition (To sort: 0, ToDo: 0, Wrong: 0)
- ▶ Coding Idiom Violation (To sort: 14, ToDo: 0, Wrong: 6)
- ▶ Style (To sort: 4, ToDo: 0, Wrong: 0)

- ❶ CriticBrowser>>#methodAdded: (Manifest-CriticBrowser)
- ❶ SelectBrowser>>#items: (Manifest-CriticBrowser)
- ❶ SelectBrowser>>#setAddAllButtonModel (Manifest-CriticBrowser)
- ❶ SelectBrowser>>#setAddButtonModel (Manifest-CriticBrowser)
- ❶ SelectBrowser>>#setRemoveAllButtonModel (Manifest-CriticBrowser)
- ❶ SelectBrowser>>#setRemoveButtonModel (Manifest-CriticBrowser)
- ❶ TApplyingOnClassSide>>#criticTheNonMetaclassClass (Traits-Kernel-Traits)
- ❶ TClassDescription>>#criticNameOn: (Traits-Kernel-Traits)



CC Workflow

No sense of system status



The screenshot shows a list of rules categorized into several groups:

- Selected Rules (To sort: 40, ToDo: 0, Wrong: 21)**
 - Bugs (To sort: 9, ToDo: 0, Wrong: 0)**
 - Debugging code left in methods (To sort: 0, ToDo: 0, Wrong: 0)
 - Messages sent but not implemented (To sort: 8, ToDo: 0, Wrong: 0)
 - Method source contains linefeeds (To sort: 0, ToDo: 0, Wrong: 0)
 - No direct access of methodDict (To sort: 1, ToDo: 0, Wrong: 0)
 - Overrides a "special" message (To sort: 0, ToDo: 0, Wrong: 0)
 - References an undeclared variable (To sort: 0, ToDo: 0, Wrong: 0)
 - Sends super new initialize (To sort: 0, ToDo: 0, Wrong: 0)
 - Sends unknown message to global (To sort: 0, ToDo: 0, Wrong: 0)
 - Subclass responsibility not defined (To sort: 0, ToDo: 0, Wrong: 0)
 - Uses True/False instead of true/false (To sort: 0, ToDo: 0, Wrong: 0)
 - Variable used, but not defined anywhere (To sort: 0, ToDo: 0, Wrong: 0)
 - Optimization (To sort: 3, ToDo: 0, Wrong: 6)**
 - Potential Bugs (To sort: 0, ToDo: 0, Wrong: 1)**
 - Design Flaws (To sort: 10, ToDo: 0, Wrong: 8)**
 - Unclassified rules (To sort: 0, ToDo: 0, Wrong: 0)**
 - Check for same statements in multiple places (To sort: 0, ToDo: 0, Wrong: 0)
 - Code Rewriting (To sort: 0, ToDo: 0, Wrong: 0)
 - Explicit requirement methods (To sort: 0, ToDo: 0, Wrong: 0)
 - Repeated method in the superclass (To sort: 0, ToDo: 0, Wrong: 0)
 - Repeated methods in the trait composition (To sort: 0, ToDo: 0, Wrong: 0)
 - Coding Idiom Violation (To sort: 14, ToDo: 0, Wrong: 0)**
 - Style (To sort: 4, ToDo: 0, Wrong: 0)**

On the right side of the screenshot, there is a list of specific code snippets with error counts:

- CriticBrowser>>#methodAdded: (Manifest-CriticBrowser)
- SelectBrowser>>#items: (Manifest-CriticBrowser)
- SelectBrowser>>#setAddAllButtonModel (Manifest-CriticBrowser)
- SelectBrowser>>#setAddButtonModel (Manifest-CriticBrowser)
- SelectBrowser>>#setRemoveAllButtonModel (Manifest-CriticBrowser)
- SelectBrowser>>#setRemoveButtonModel (Manifest-CriticBrowser)
- Applying...Side>>#critical...Meta...Traits (Traits-Kernel-Traits)
- ...>#critical...Traits

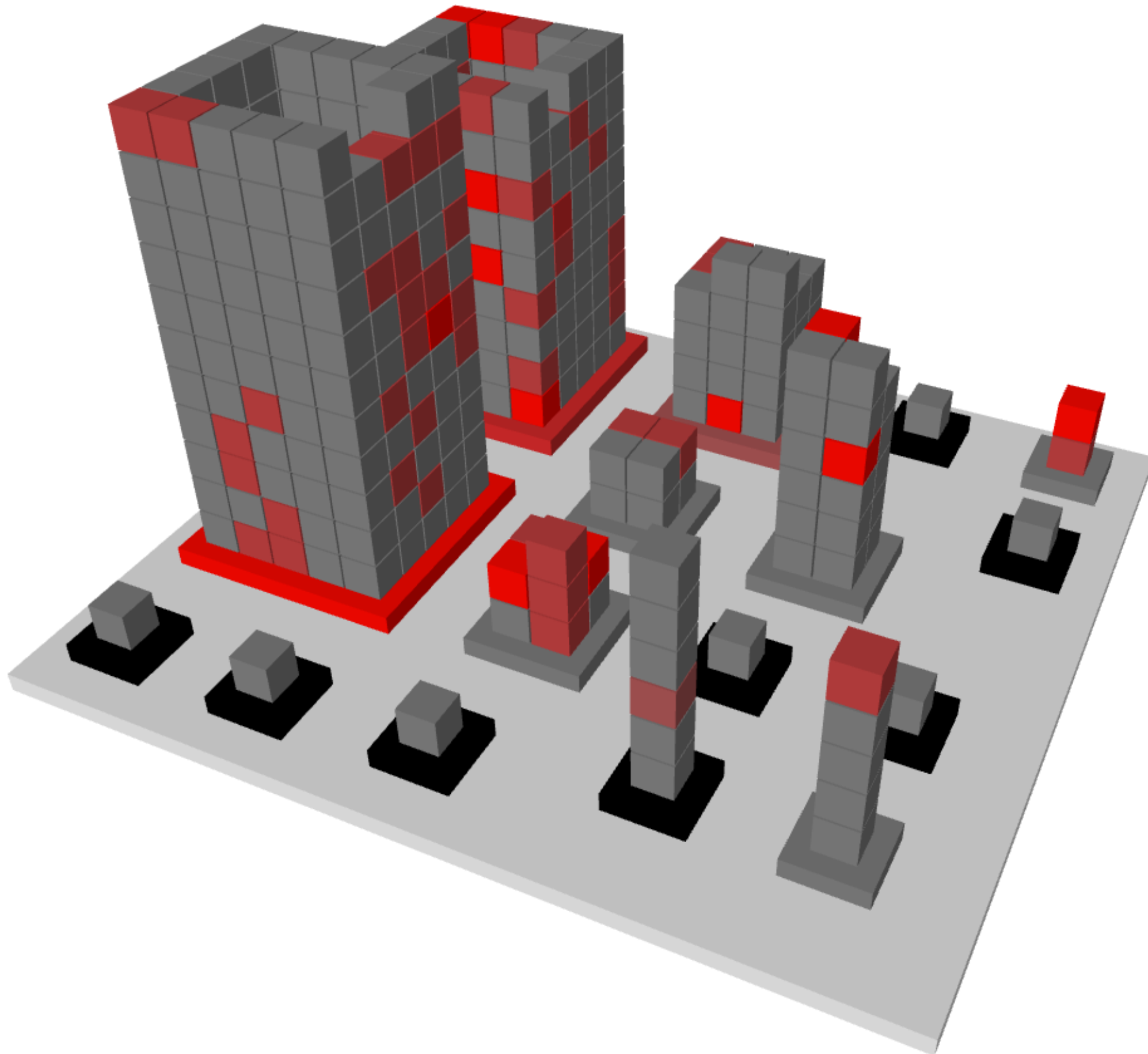


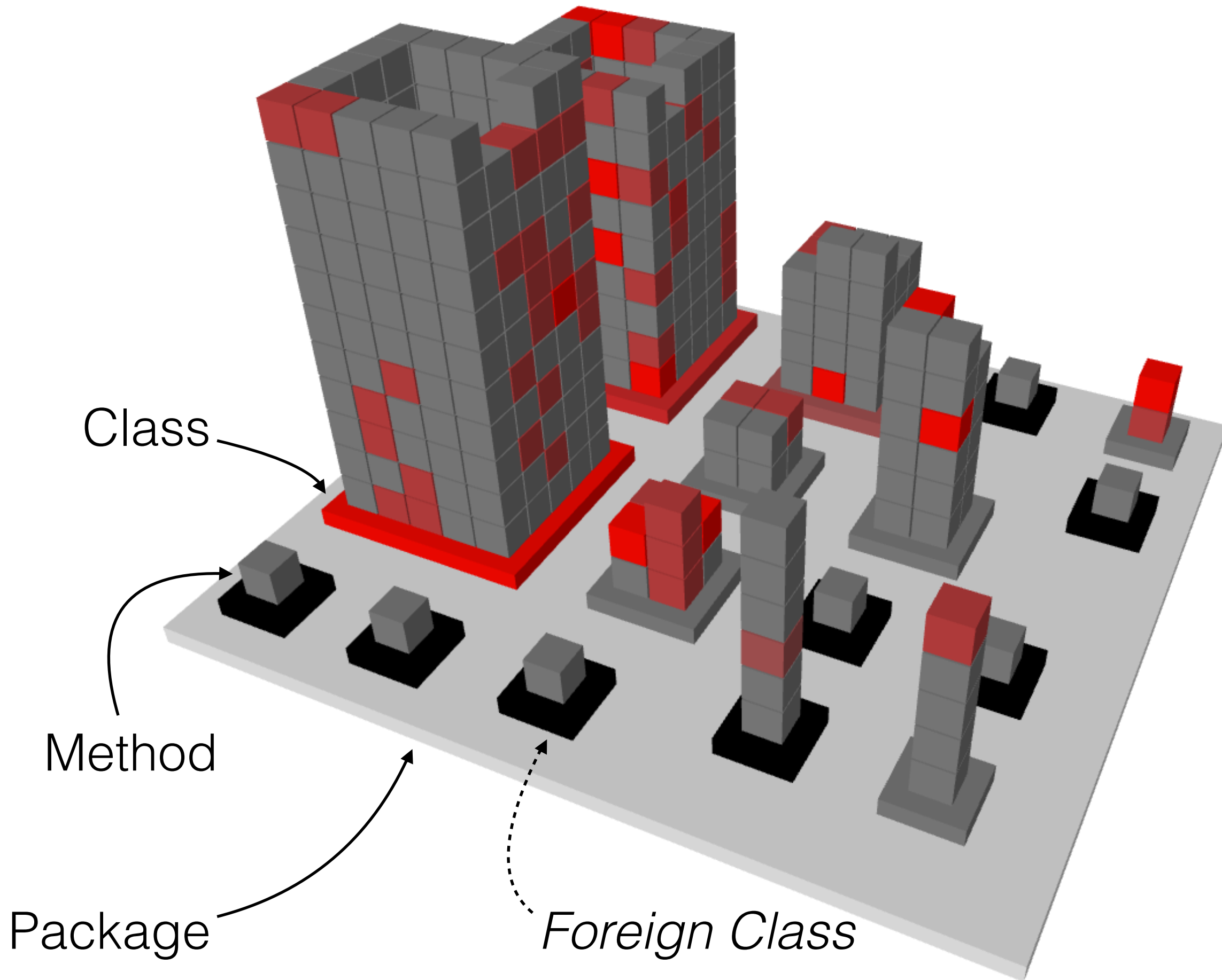
CC Workflow

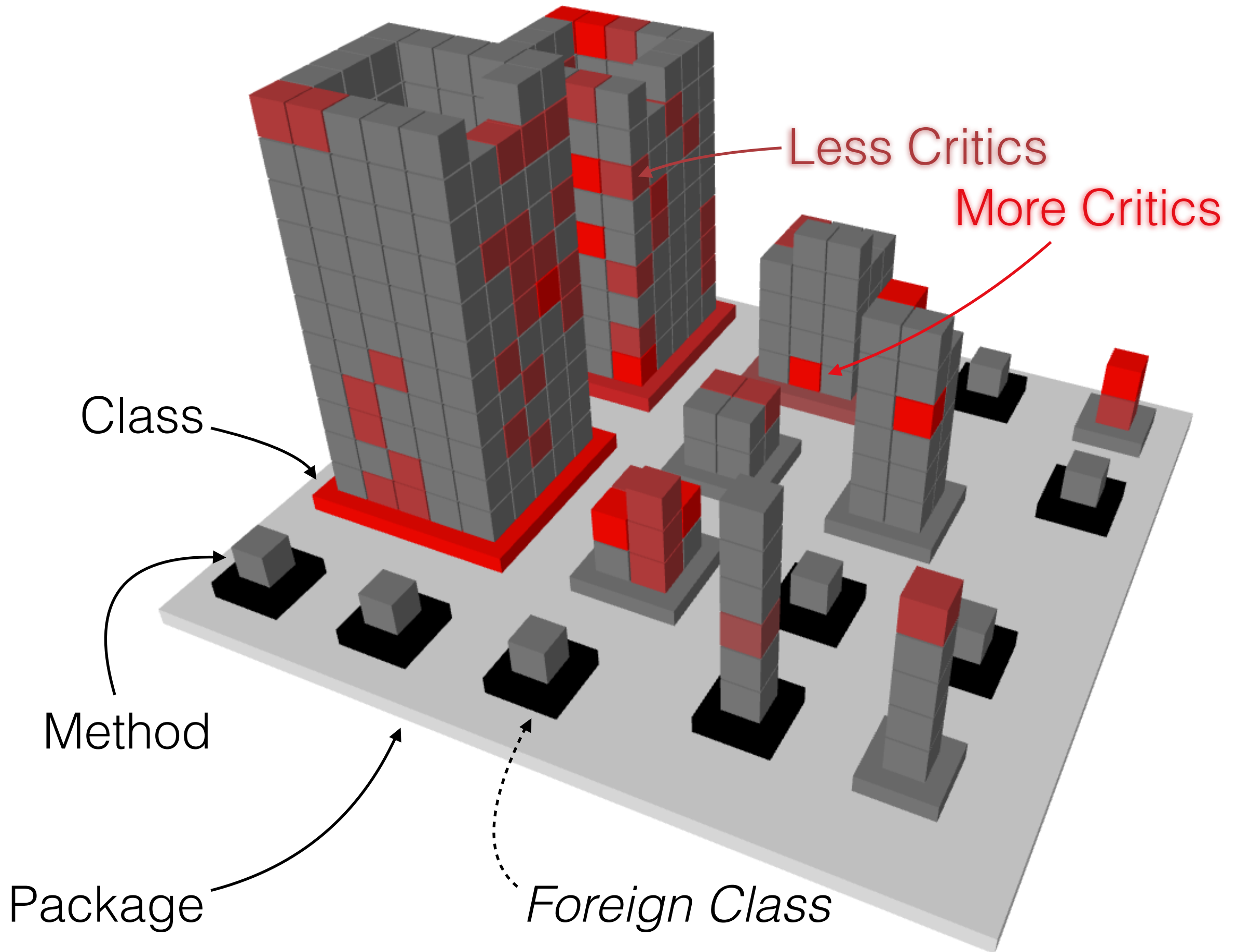
No sense of system status

Breaks flow

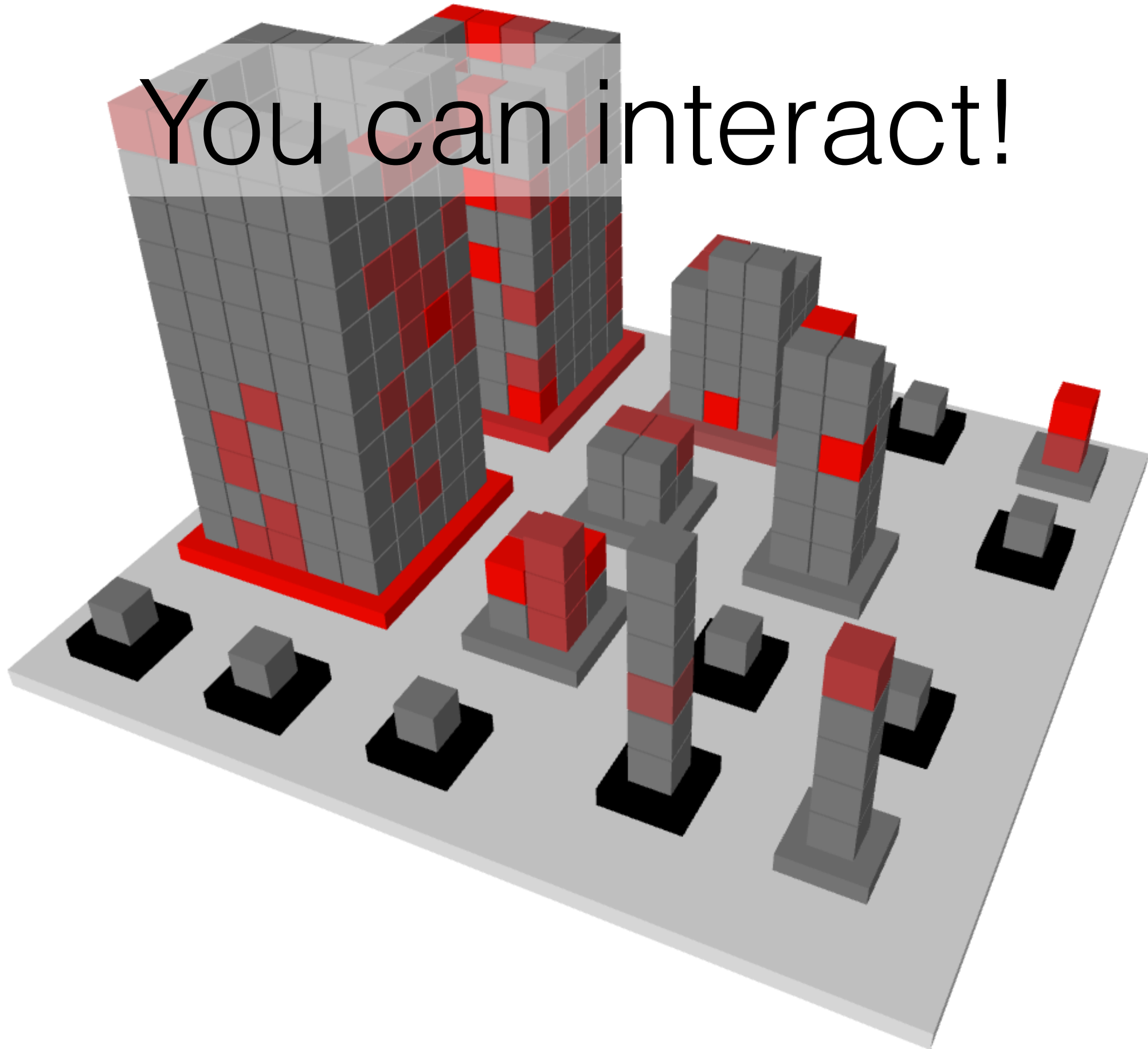


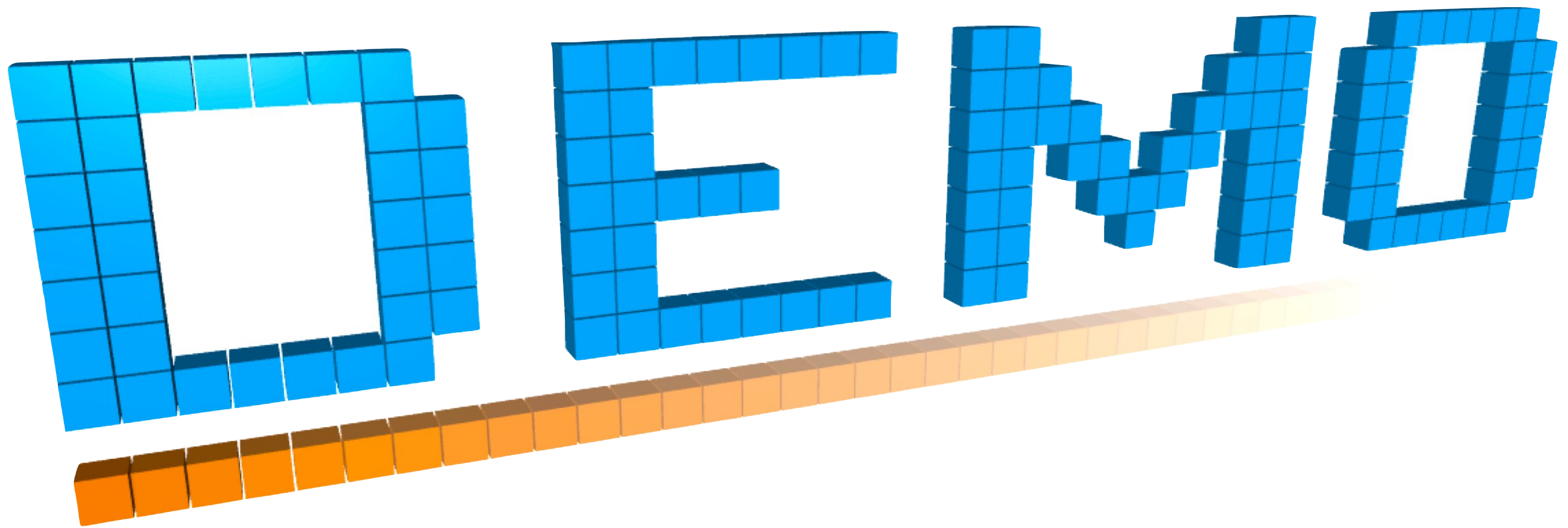




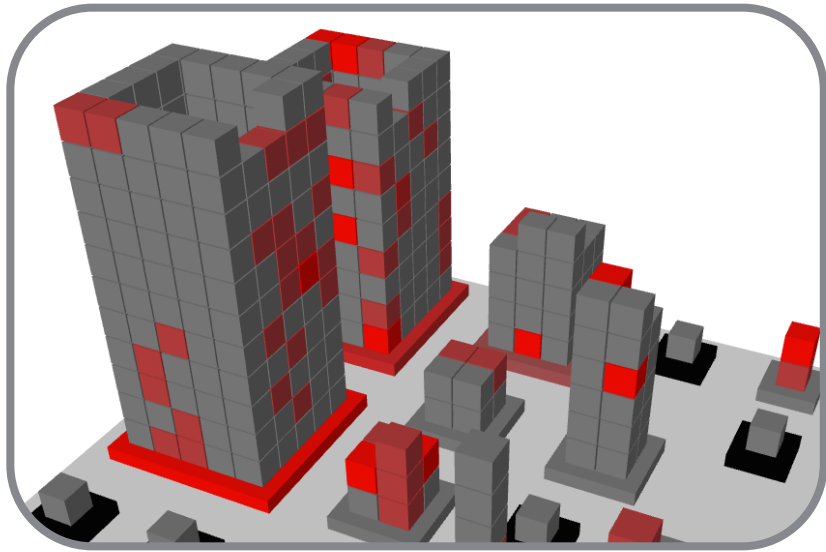


You can interact!

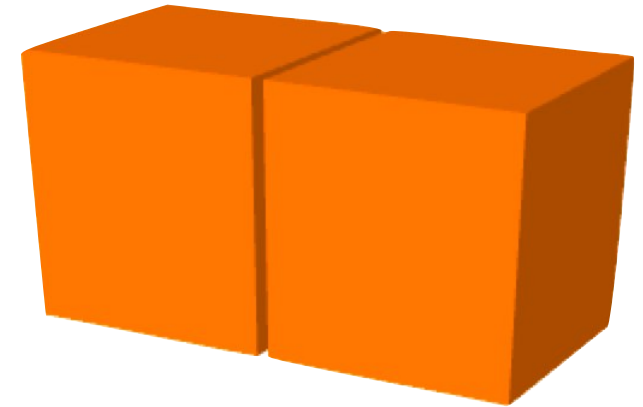




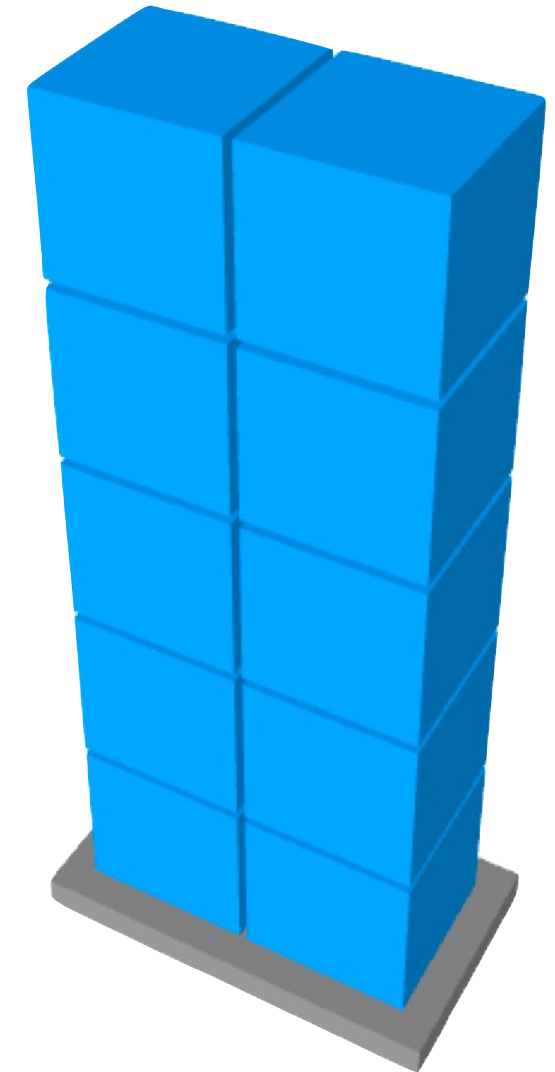
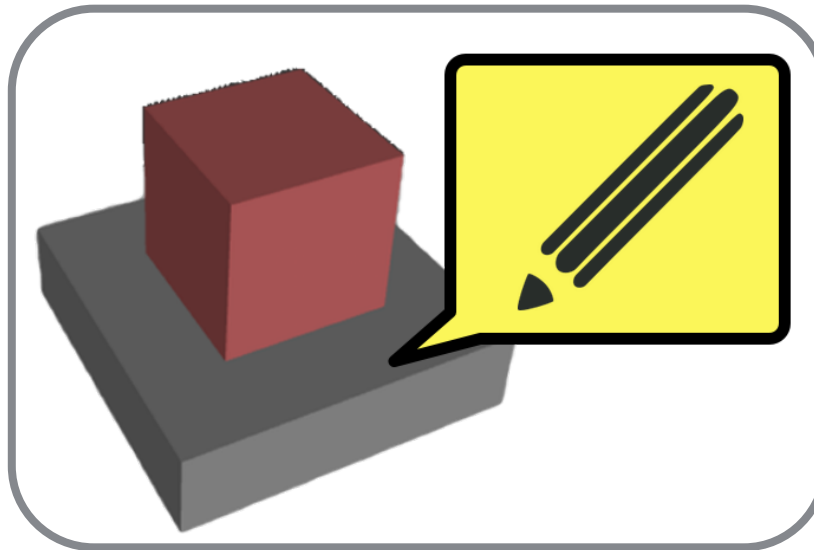
* this viz is made with Roassal3D



```
object: receiver performMessage:  
"Check if the message to perform  
| selector arguments lookupClass  
  
selector := aMessage arguments first  
arguments := aMessage arguments second  
lookupClass := aMessage arguments third  
  
selector isSymbol  
  ifFalse: [^self reflectiveFailedToken].  
selector numArgs = arguments size  
  ifFalse: [^self reflectiveFailedToken].  
(receiver isKindOfClass: lookupClass)  
  ifFalse: [^self reflectiveFailedToken].  
  
message := Message selector: selector arguments: arguments  
to: receiver class: lookupClass  
  
^self send: message to: receiver
```



```
messageInSuperclass: aMessage  
to perform is valid and if so,  
perform it.  
| lookupClass message|  
lookupClass := aMessage arguments first.  
message arguments second.  
message arguments third.  
  
lookupClass reflectiveFailedToken].  
arguments size  
lookupClass reflectiveFailedToken].  
lookupClass := lookupClass)  
lookupClass reflectiveFailedToken].  
  
selector: selector arguments: arguments  
to: receiver class: lookupClass
```



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