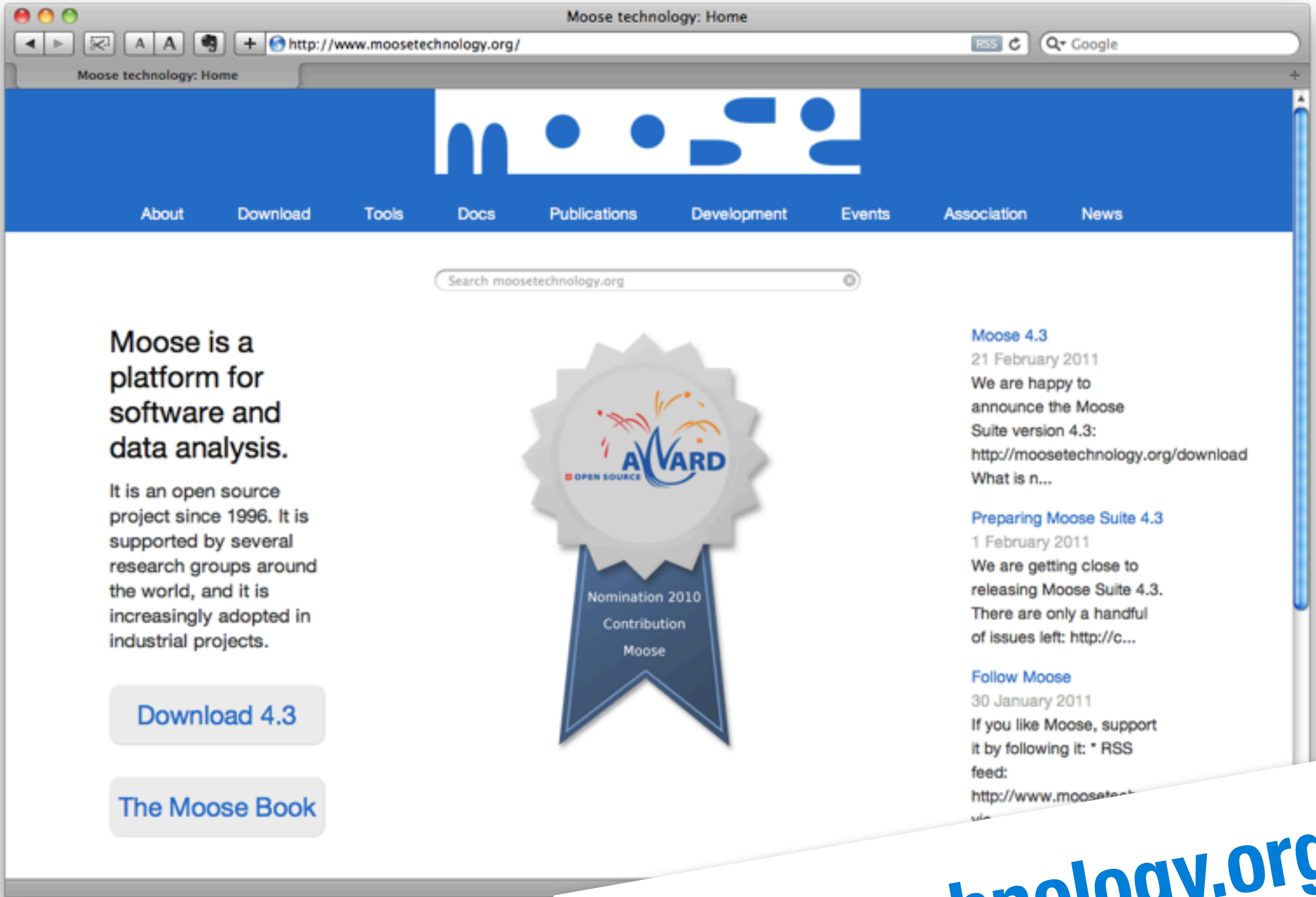
Lumane assessment with Moose

Stéphane Ducasse
clearVue
INRIA



moosetechnology.org

Humane/agile assessment: Home

http://humane-assessment.com/

Google

Humane/agile assessment: Home

Humane assessment

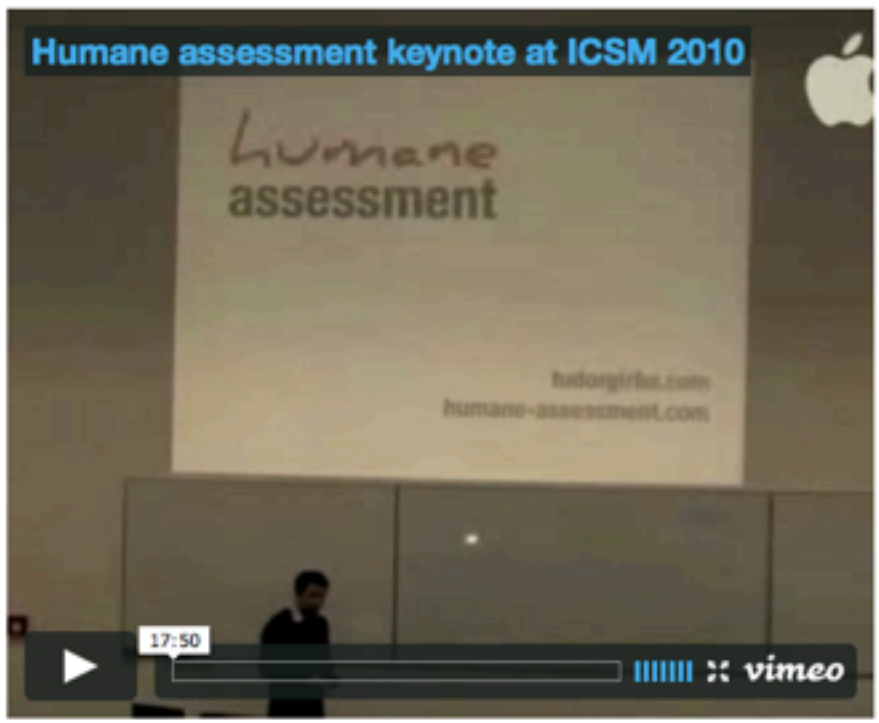
Assessment is an act of reasoning. It is what we should do before taking a decision. However, when large amounts of data are involved, assessment gets complicated because we are just not equipped to handle many details.

Data is here to stay. Whether under the form of software systems or otherwise, we will only get more, so we better get good at dealing with it. We need tools to deal with the vastness of data, but these tools should bend around humans, rather than requiring humans to conform to them. Moose is such a tool. It offers an extensive, flexible and open-source platform for analyzing data in general and software systems in particular.

Tools are important, but we need to recognize that assessment is a human activity, and that we have to tackle it as such. Analysis tools should merely assist analysts by crunching data only when and how it is required.

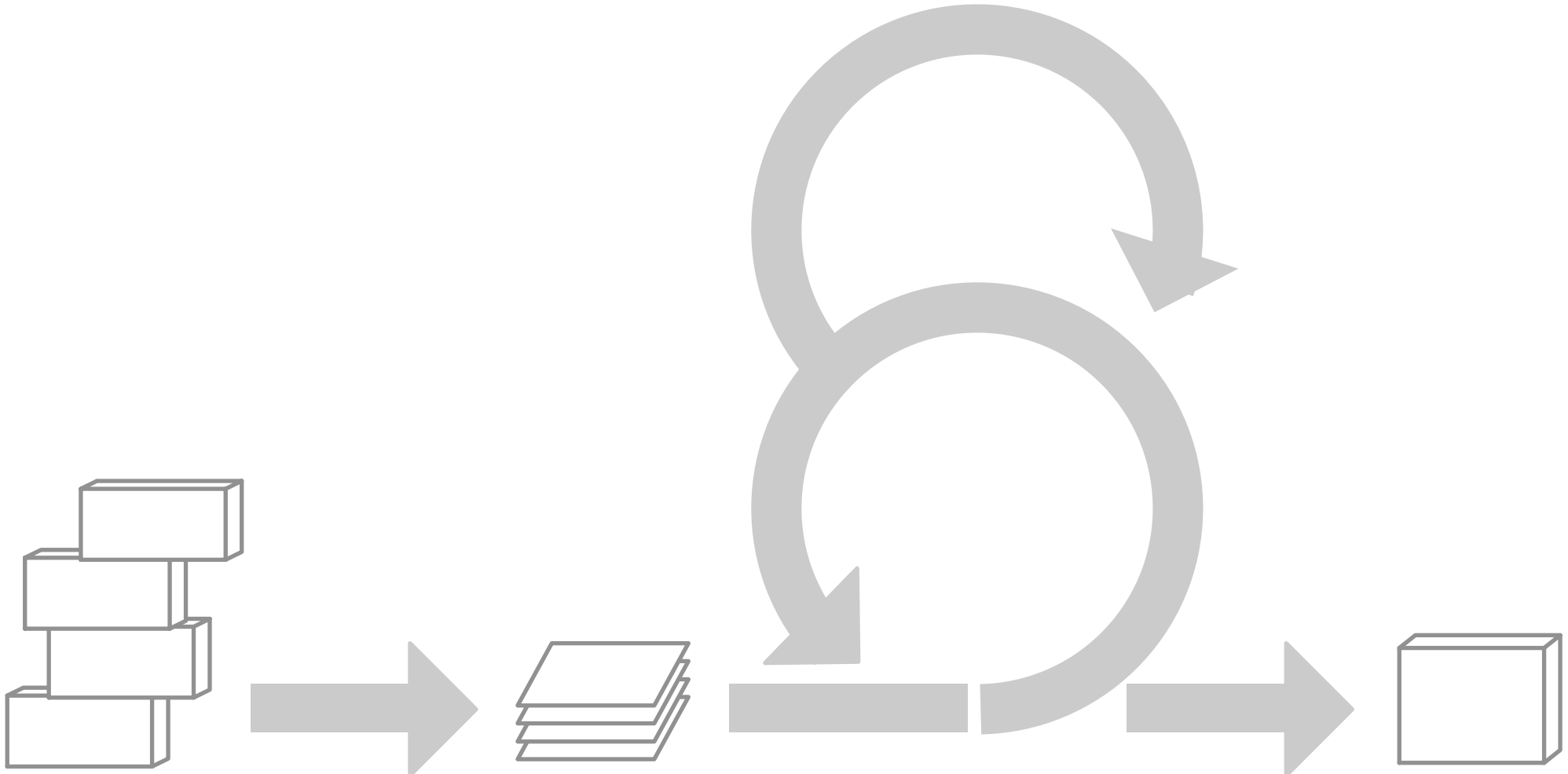
I call this approach *the humane (or agile) assessment*. More information can be found by watching the short keynote to the right, by reading the humane assessment booklet, or by contacting me.

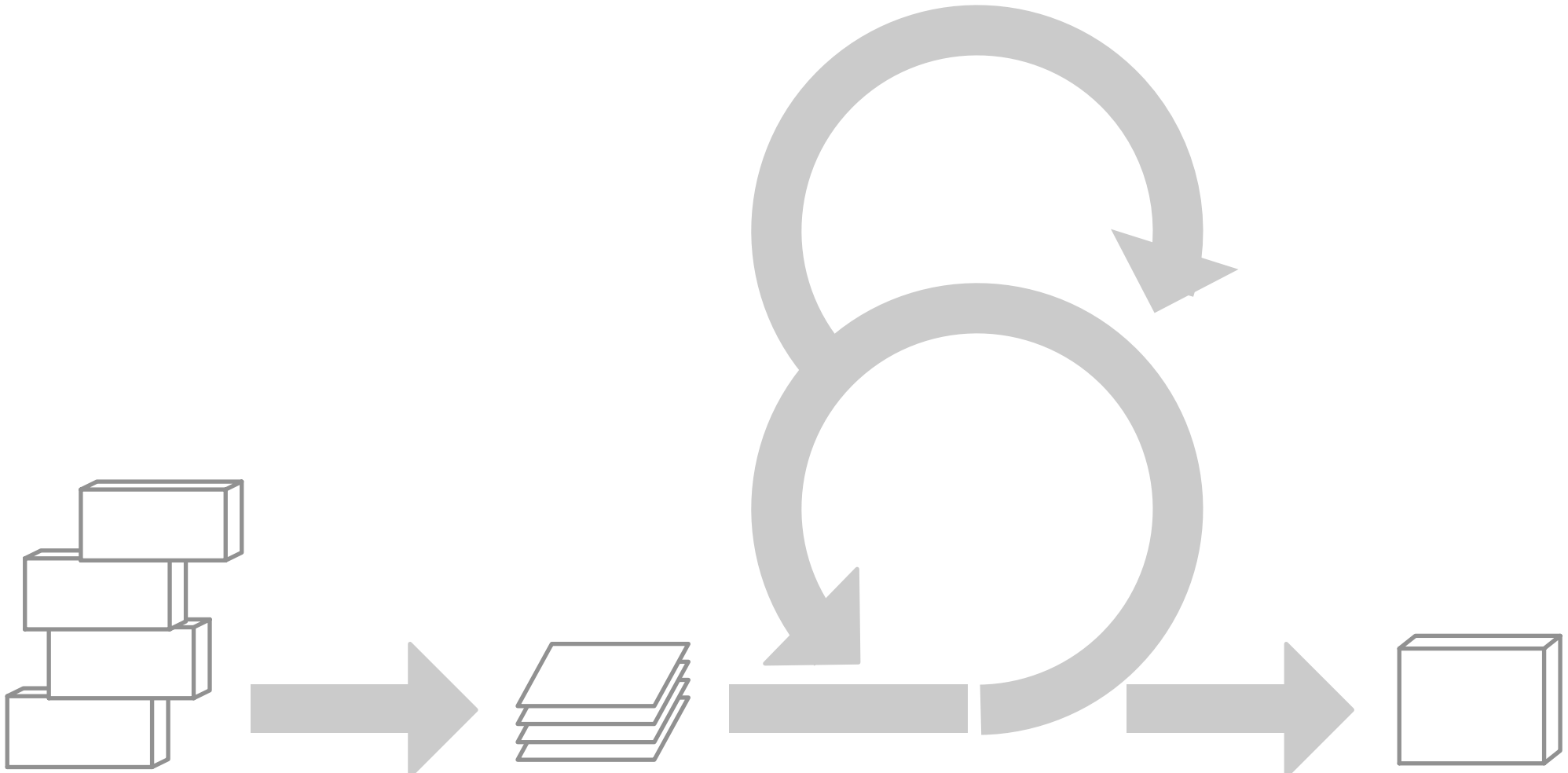
Happy assessment,
Tudor Girba



Download the booklet

humane-assesment.com

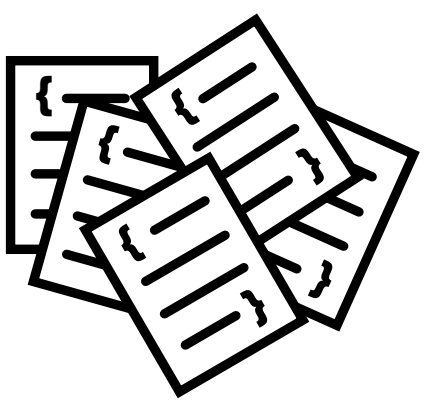




feedback is key

feedback is key

contextual
continuous
feedback **is key**



= *complex*, **large**

250’000 lines of code

250’000 lines of code

*** 2 = 500’000 seconds**

250’000 lines of code

* 2 = 500’000 seconds

/ 3600 ~ 140 hours

250’000 lines of code

* 2 = 500’000 seconds

/ 3600 ~ 140 hours

/ 8 ~ 18 days

250’000 lines of code

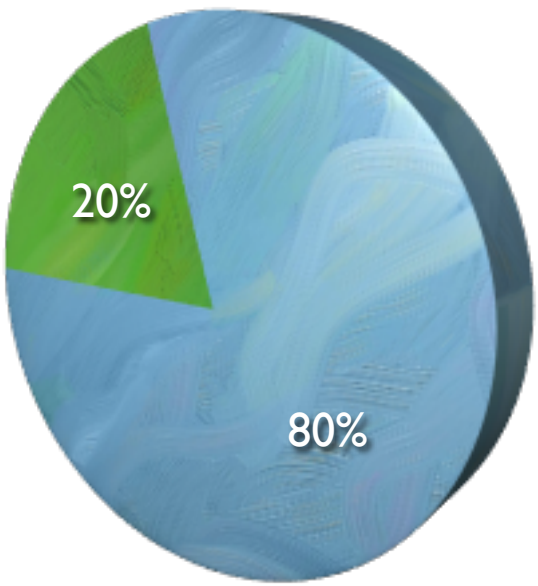
* 2 = 500’000 seconds

/ 3600 ~ 140 hours

/ 8 ~ 18 days

/ 20 ~ **1 month**

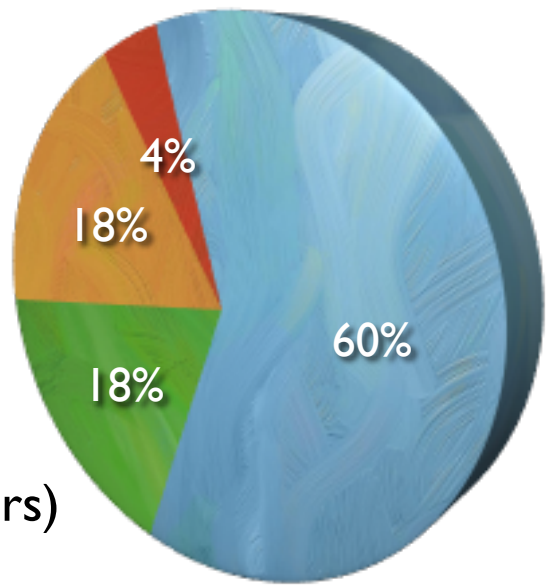
Maintenance is *continuous* Development



Between **50**% and **80**% of ***global*** effort is spent on "maintenance" !

4% Other

18% Adaptive
(new platforms)



18% Corrective
(fixing reported errors)

60% Perfective
(new functionality)

"Maintenance"

Software is like geranium...

when dead, it does not bloom!

Software is a living entity...

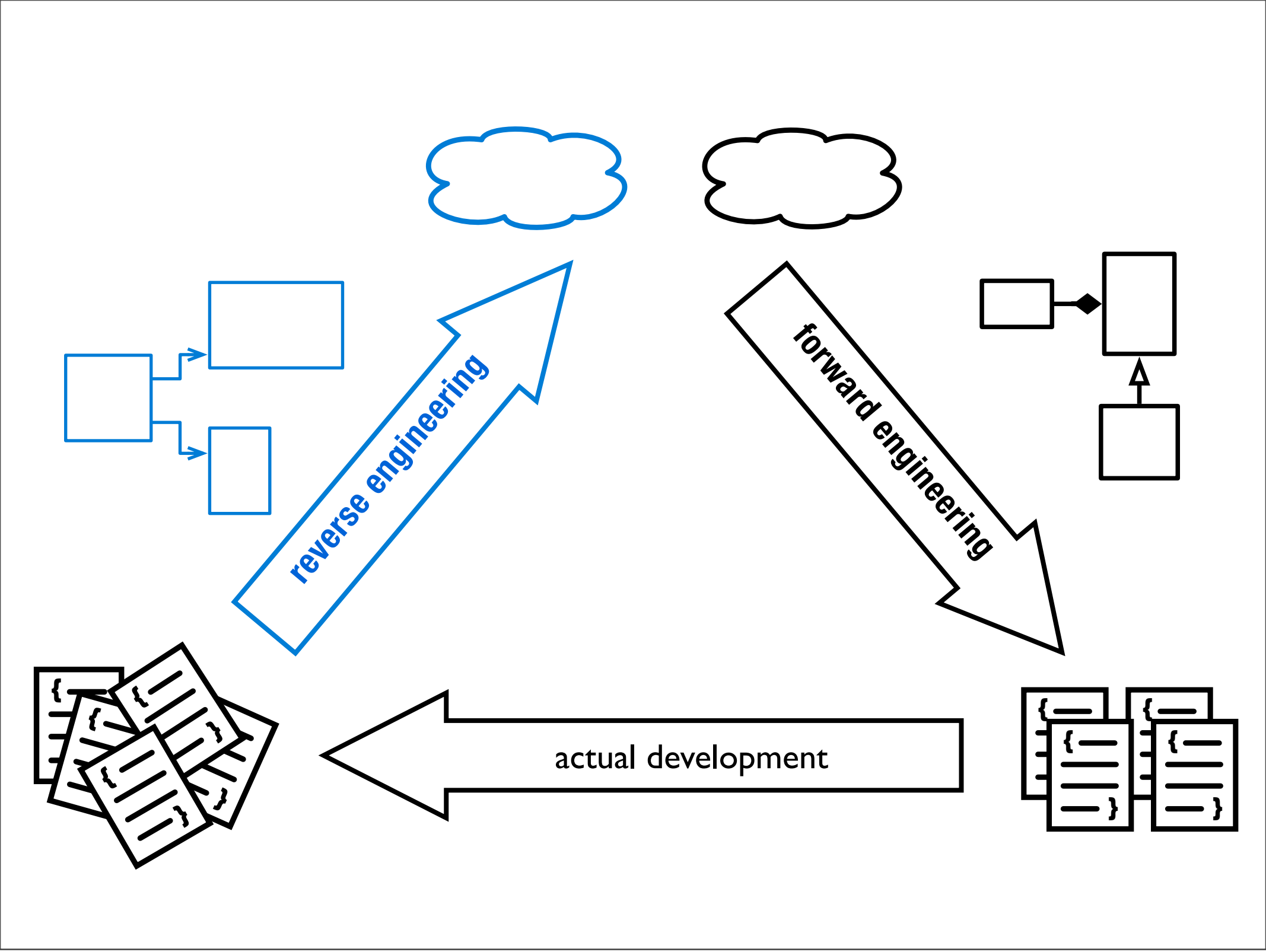
Early decisions were certainly good at that time

But the context *changes*

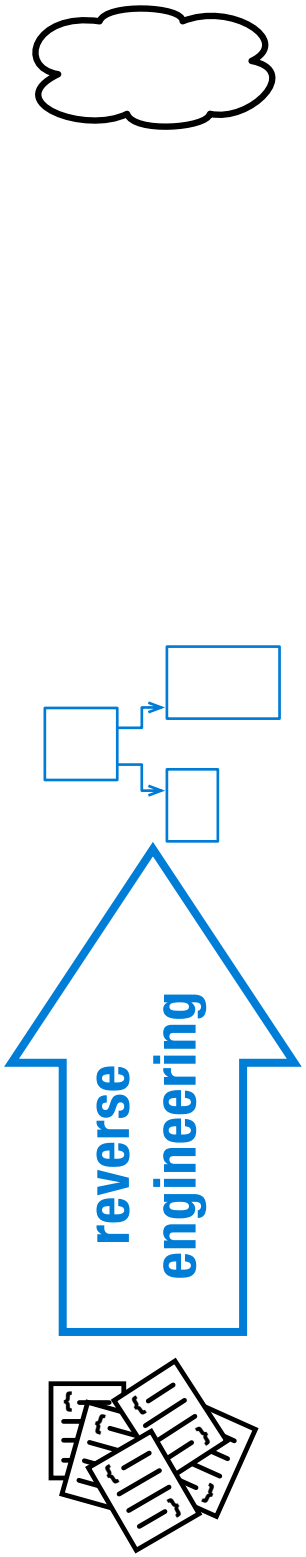
Customers *change*

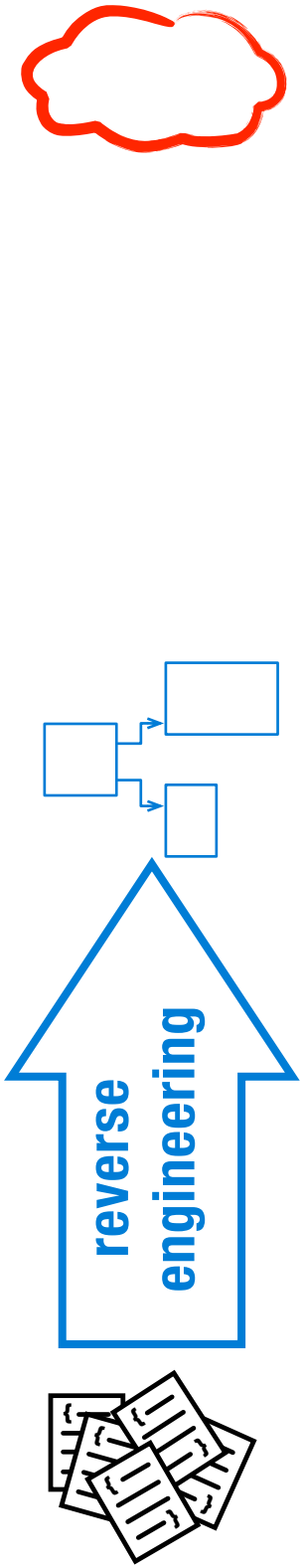
Technology *changes*

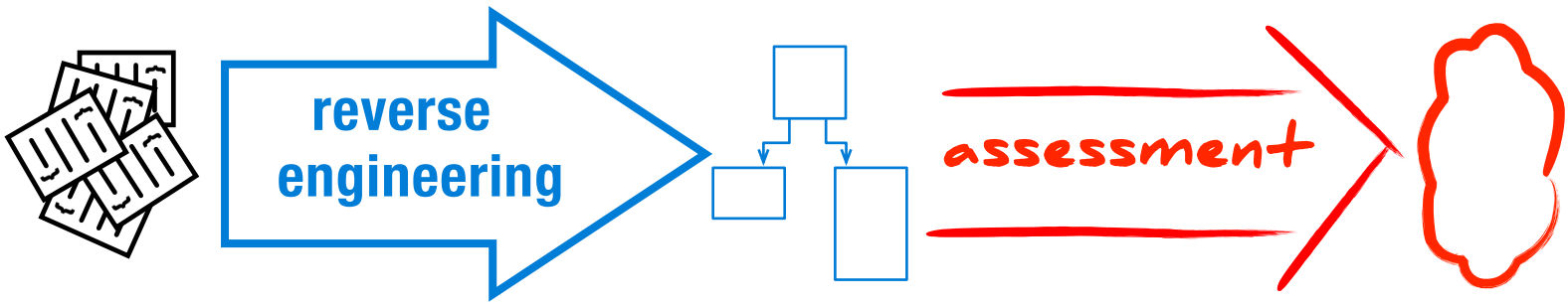
People *change*

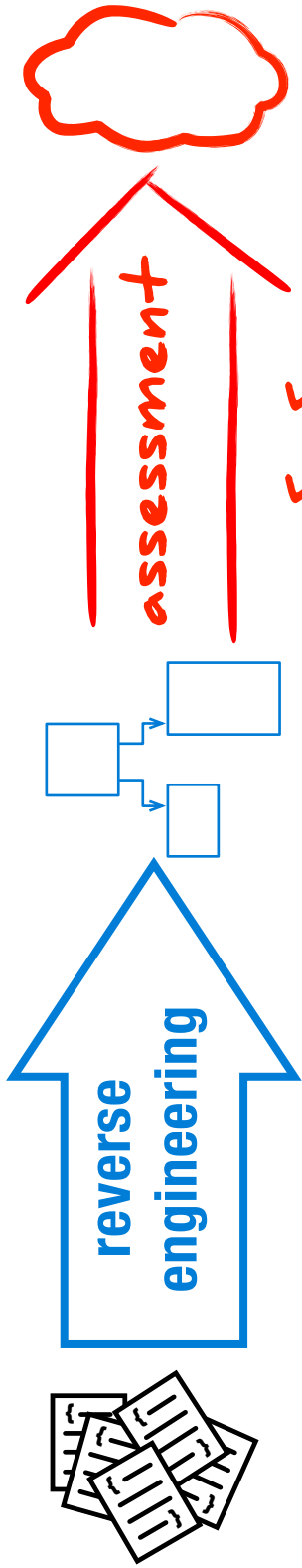

reverse engineering
forward engineering
actual development


reverse
engineering

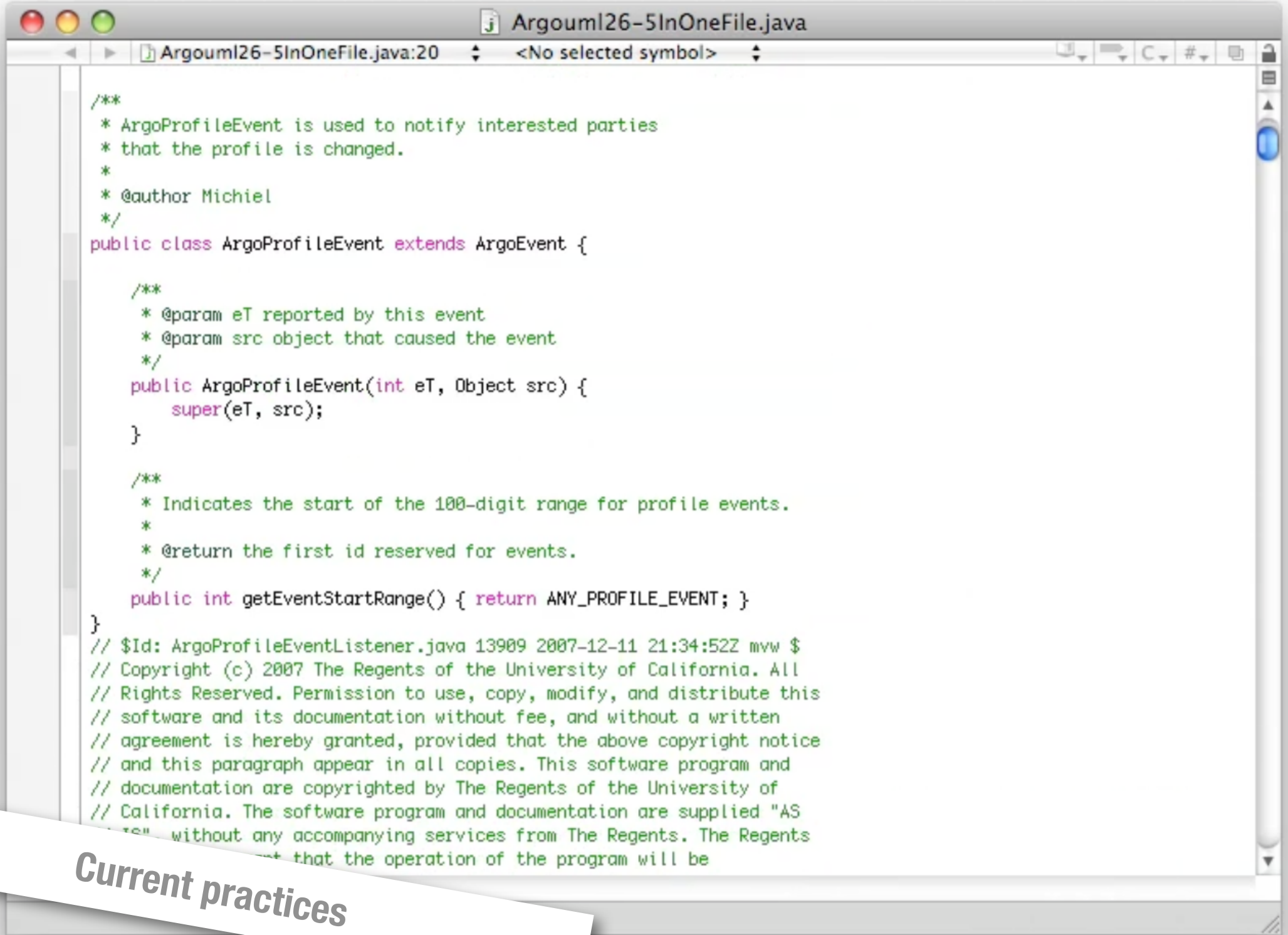

reverse
engineering


reverse
engineering

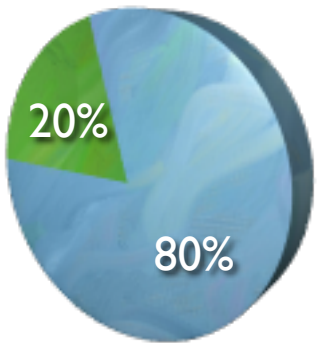

reverse
engineering
assessment

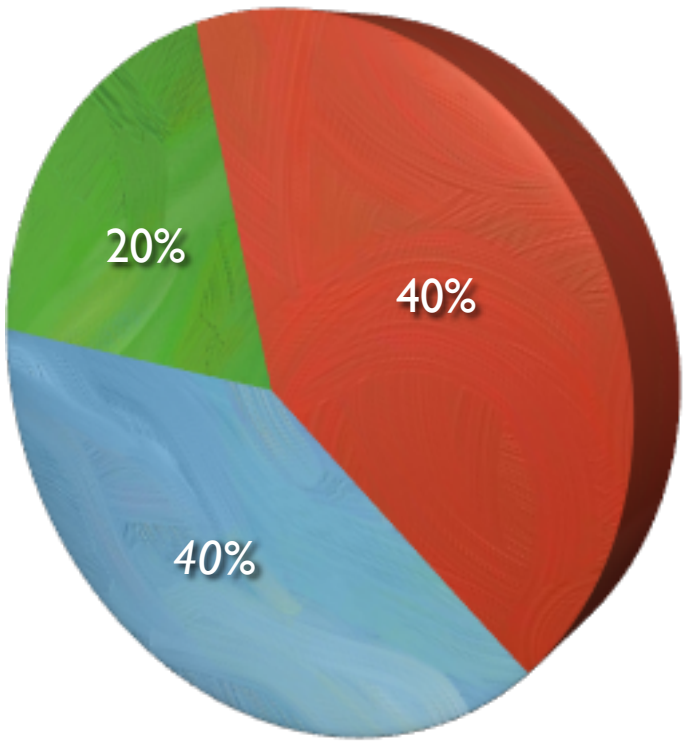


what is the current situation?
what can we do about it?


Argouml26-5InOneFile.java
Argouml26-5InOneFile.java:20
<No selected symbol>
/**
 * ArgoProfileEvent is used to notify interested parties
 * that the profile is changed.
 *
 * @author Michiel
 */
public class ArgoProfileEvent extends ArgoEvent {
/**
 * @param eT reported by this event
 * @param src object that caused the event
 */
public ArgoProfileEvent(int eT, Object src) {
super(eT, src);
}
/**
 * Indicates the start of the 100-digit range for profile events.
 *
 * @return the first id reserved for events.
 */
public int getEventStartRange() { return ANY_PROFILE_EVENT; }
}
// $Id: ArgoProfileEventListener.java 13909 2007-12-11 21:34:52Z mvw $
// Copyright (c) 2007 The Regents of the University of California. All
// Rights Reserved. Permission to use, copy, modify, and distribute this
// software and its documentation without fee, and without a written
// agreement is hereby granted, provided that the above copyright notice
// and this paragraph appear in all copies. This software program and
// documentation are copyrighted by The Regents of the University of
// California. The software program and documentation are supplied "AS
// without any accompanying services from The Regents. The Regents
that the operation of the program will be
Current practices

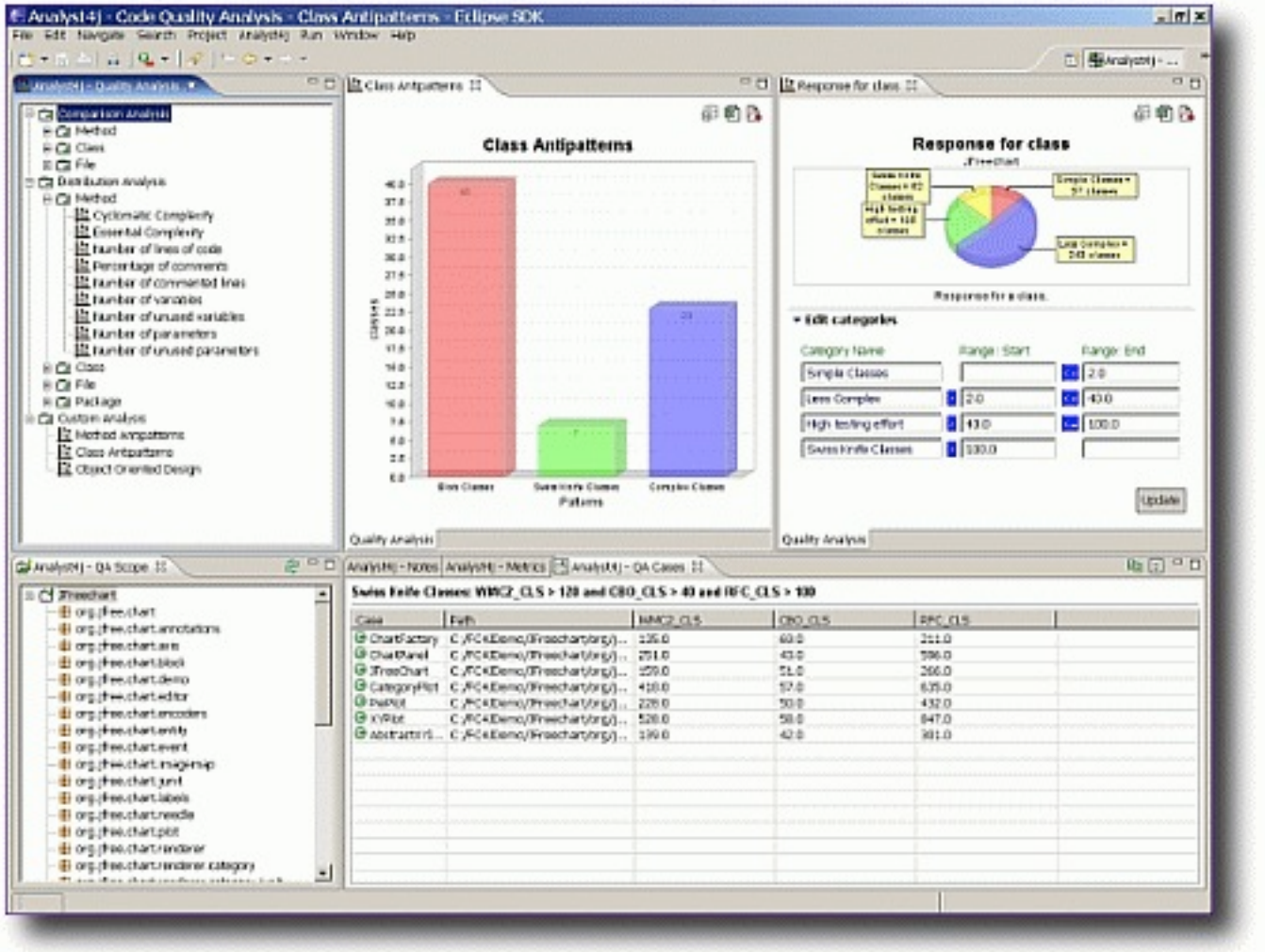
50% of maintenance is spent reading code!





Between **50**% and **80**% of ***global*** effort is spent on "maintenance" !

We spend a lot of money at the wrong place with the wrong tools!



devMetrics Project Report

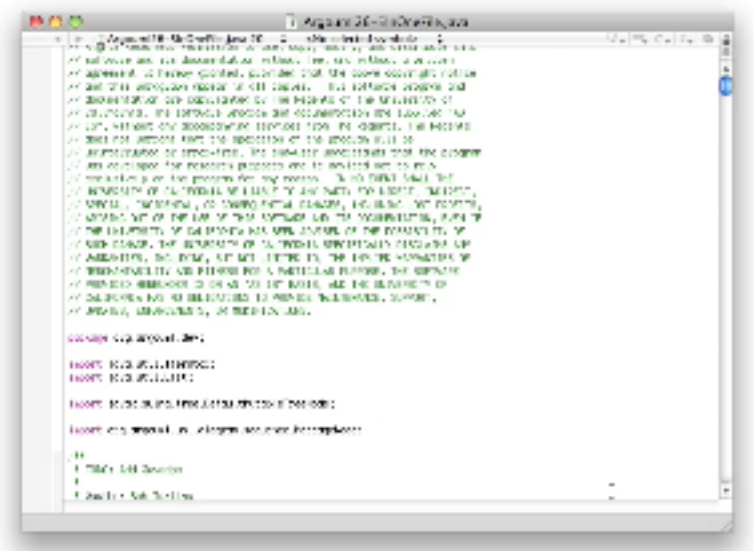
Run at: 2:26:23 PM, Wednesday, May 26, 2004

Project Elements						Comments		Code Complexity		Statements/Member		Members/Class	
Project	Files	Classes	Members	Lines	Statements	Count	Density	Avg	Max	Avg	Max	Avg	Max
ApplicationStorage	3	2	20	352	120	59	16.8%	2.4	6	10.9	54	10.0	14
RSSConnect	13	15	465	7,879	4,634	895	11.4%	4.9	34	21.7	573	31.0	229
RichTextBoxSupportsXHTML	6	5	41	1,073	506	136	12.7%	12.1	111	19.5	228	8.2	13
XHTML Displaying RichTextBox	2	1	13	264	95	49	18.6%	2.0	3	15.8	63	13.0	13
	24	**23**	**539**	**9,568**	**5,355**	**1,139**	**11.9%**	**5.4**	**111**	**20.8**	**573**	**23.4**	**229**

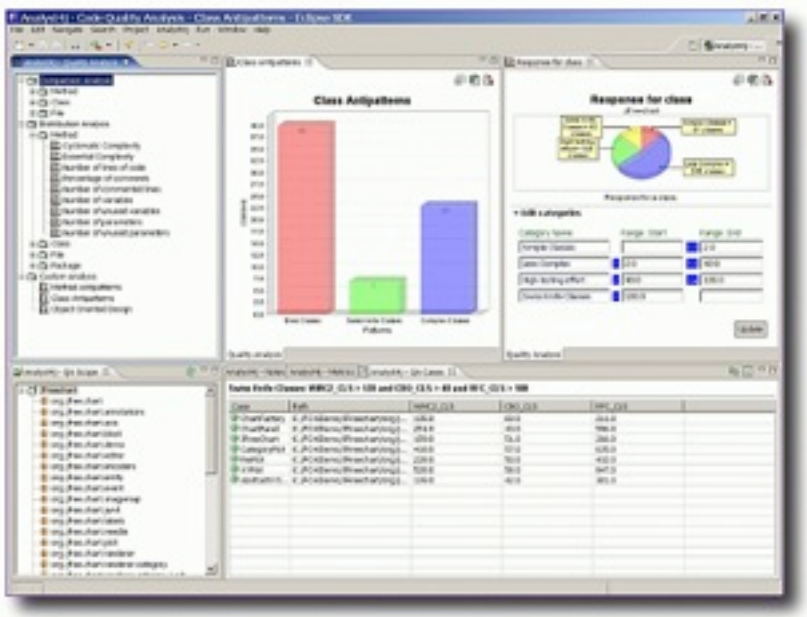
By Anticipating Minds, Inc.

Current practices

manual
tailored

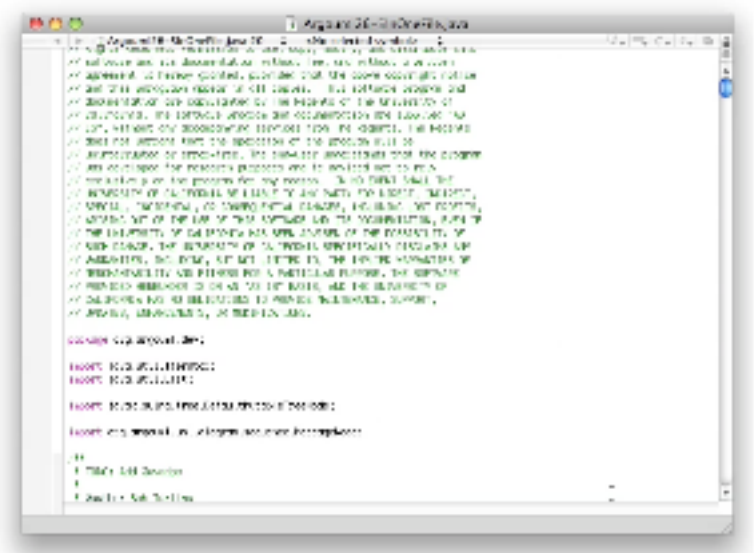


automatic
generic



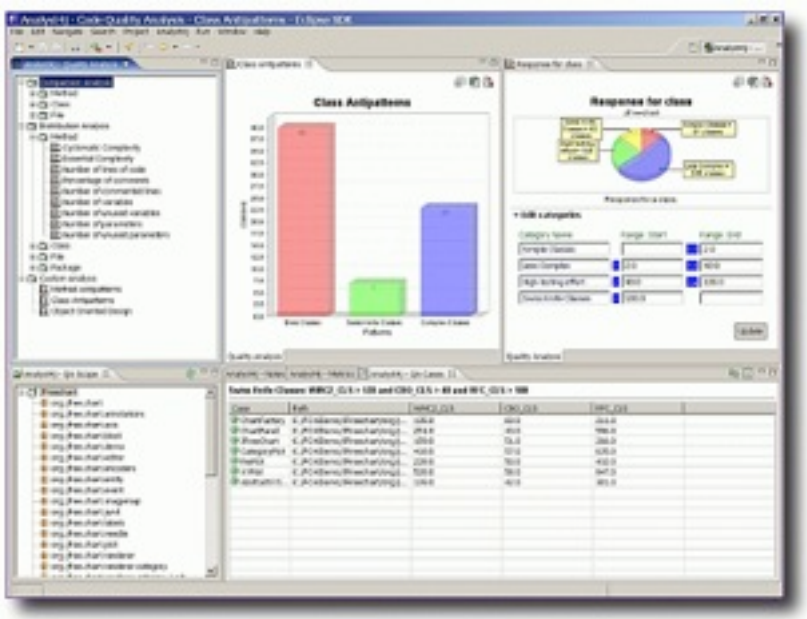
~~manual~~

tailored



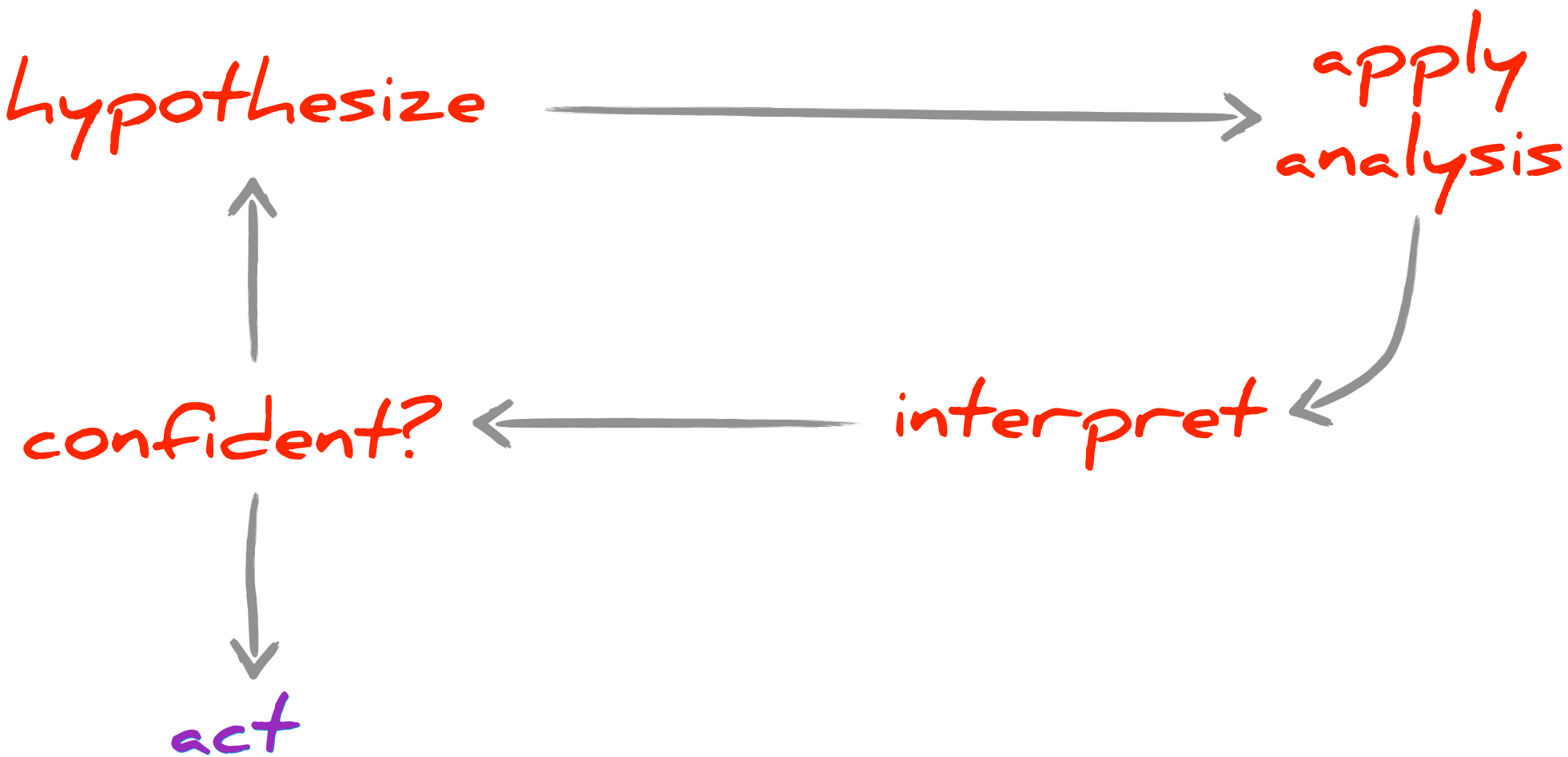
automatic

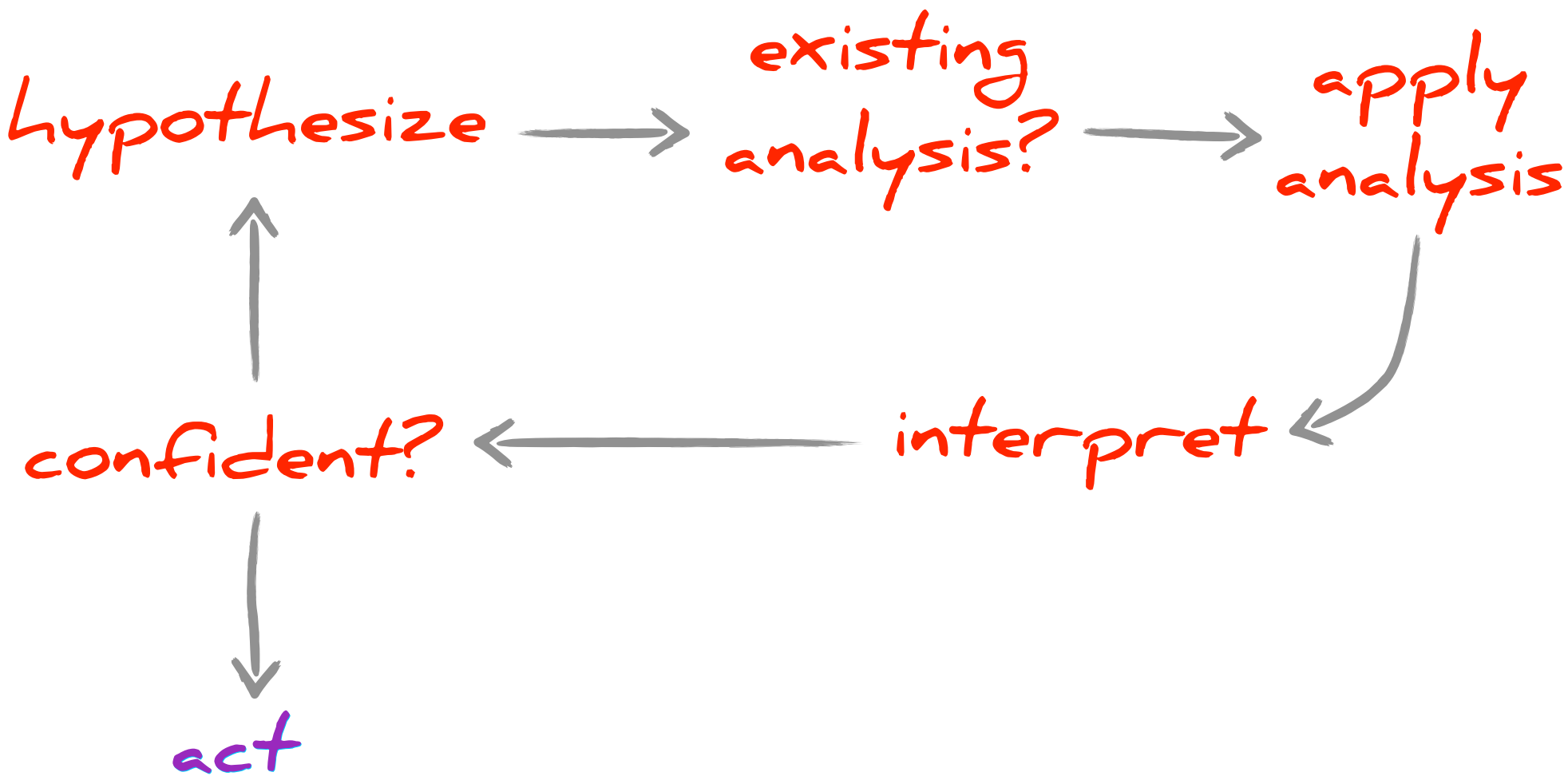
~~generic~~

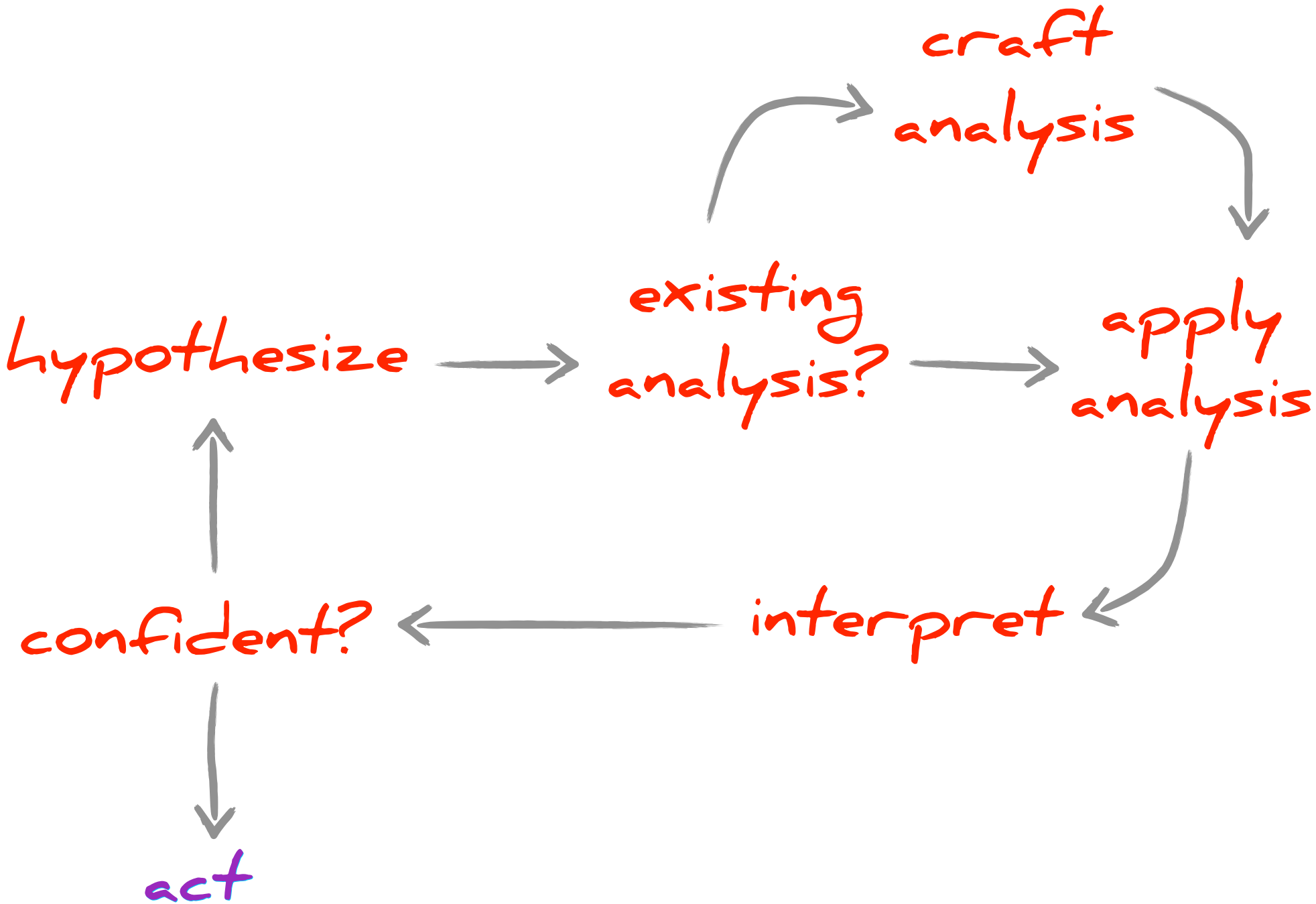


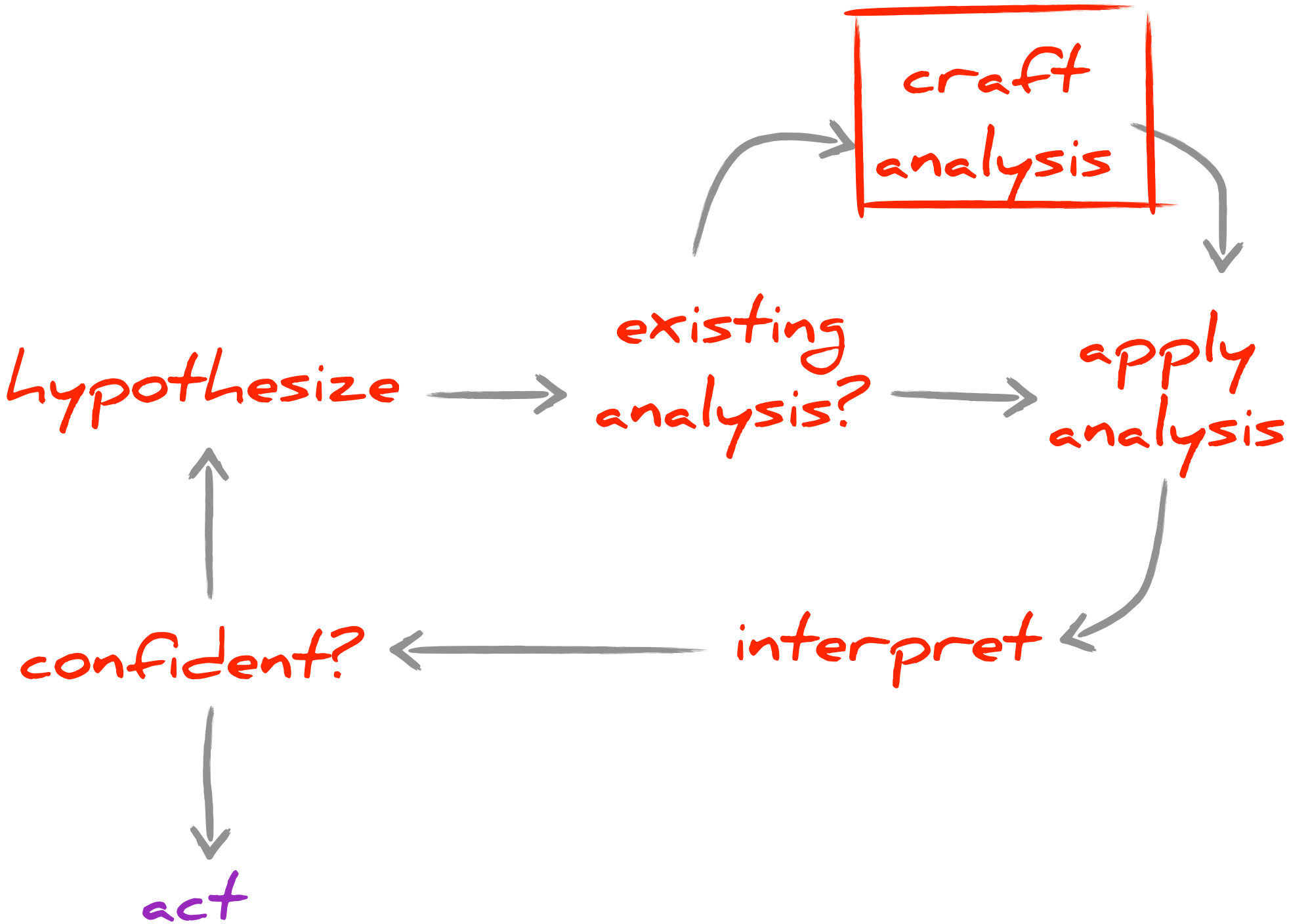
We need dedicated tools to *help* us!

We need customable tools to *control and give feedback*

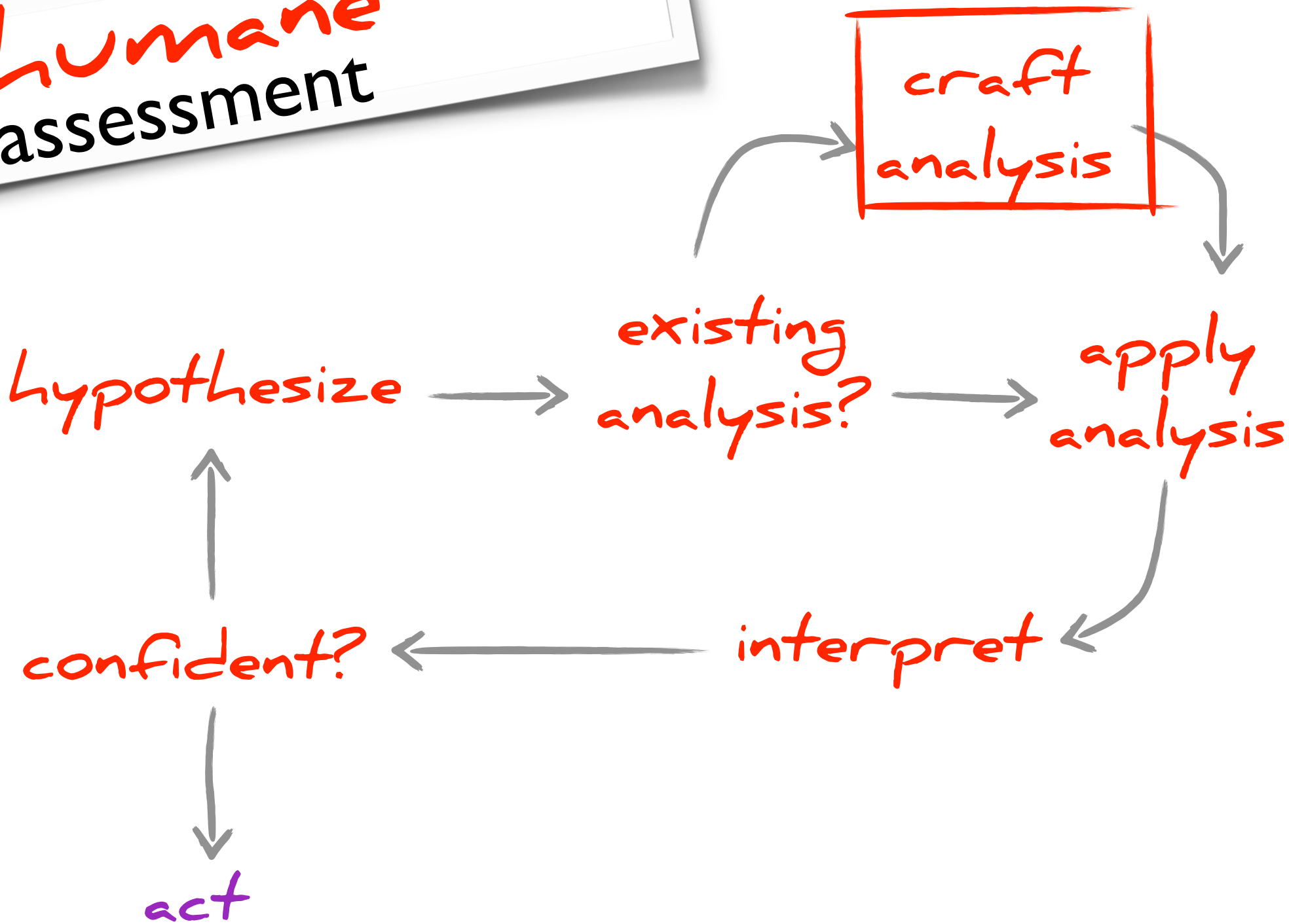

hypothesize
apply analysis
interpret
confident?
act


hypothesize
existing analysis?
apply analysis
interpret
confident?
act

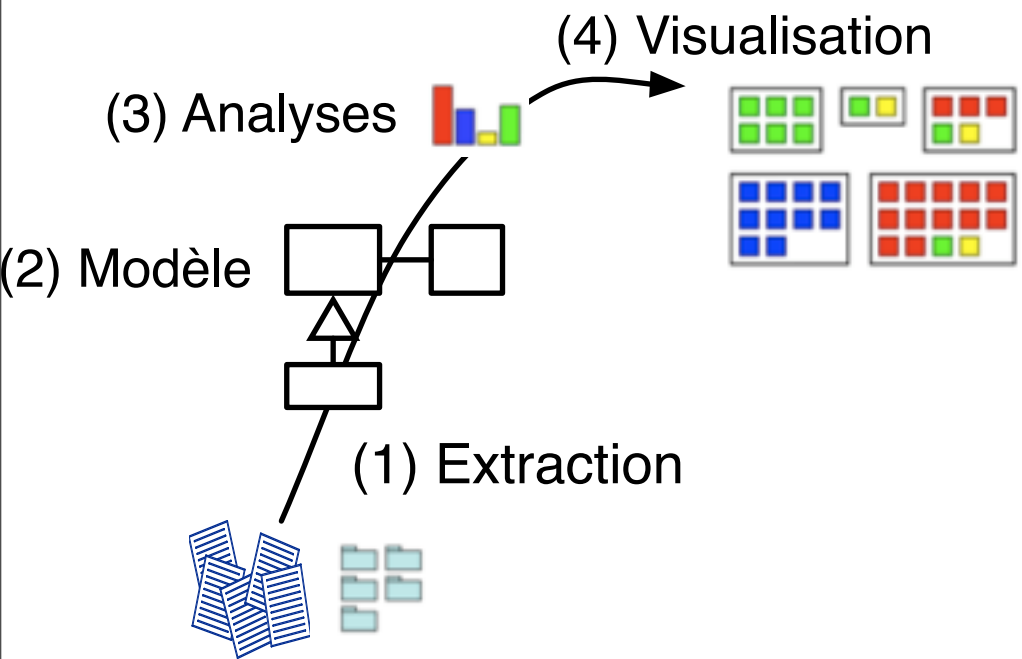

craft
analysis
existing
analysis?
apply
analysis
hypothesize
interpret
confident?
act


craft analysis
existing analysis?
apply analysis
hypothesize
confident?
interpret
act

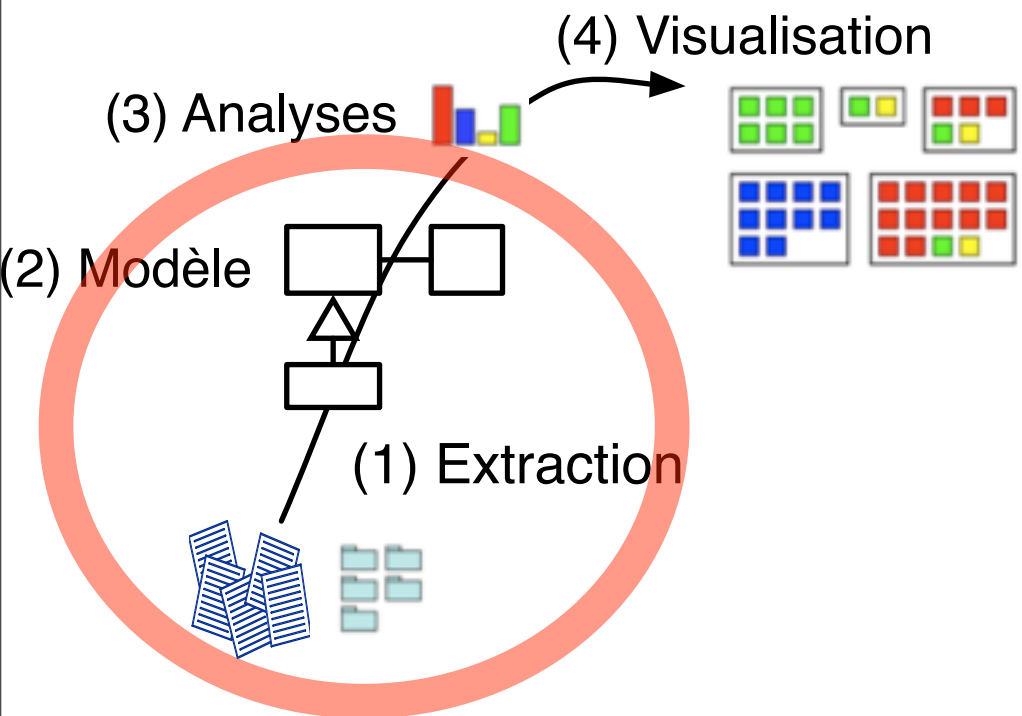
Lumane assessment



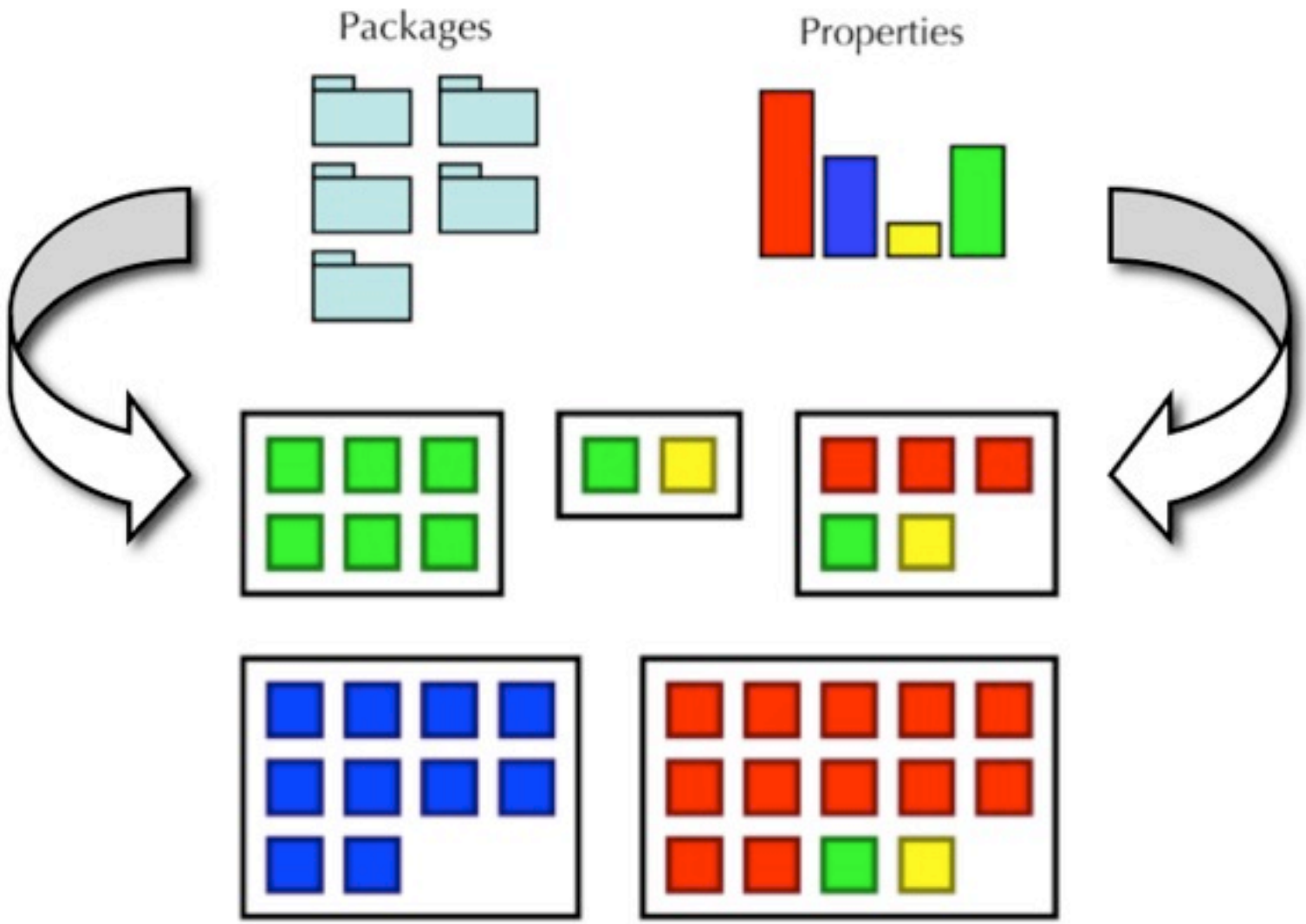
An example: who is really behind that package?



(2) Define a model for code ownership

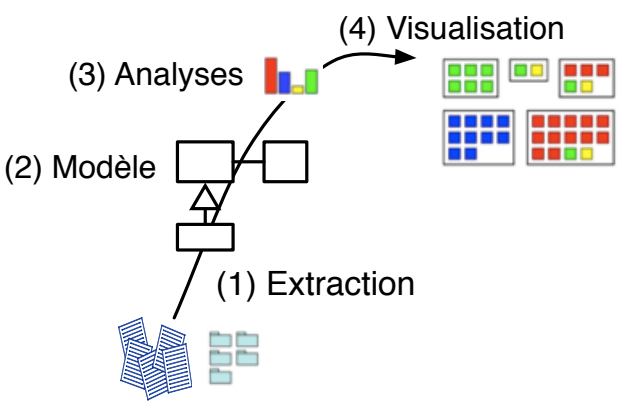


(3) Use adequate Map



Distribution Map

shows properties over structure

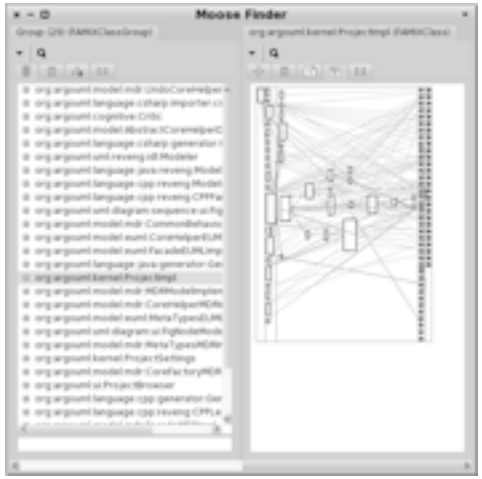


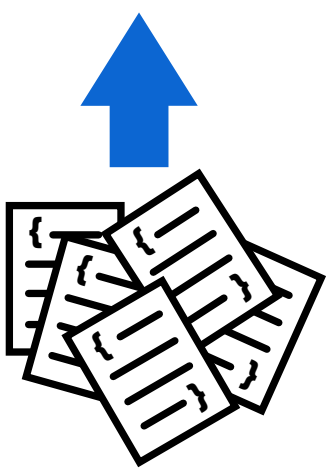
Ducasse et al 2006

classes select: #isGod

McCabe = 21

LOC = 753,000

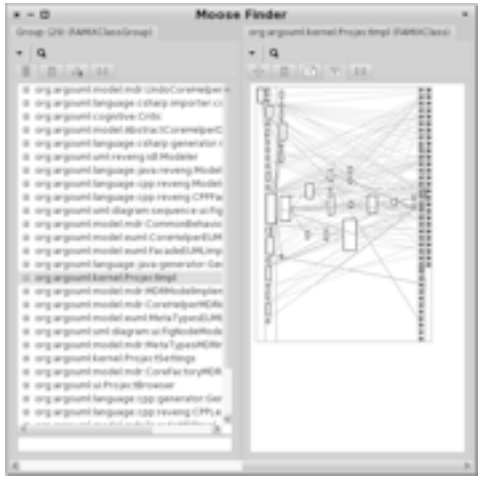


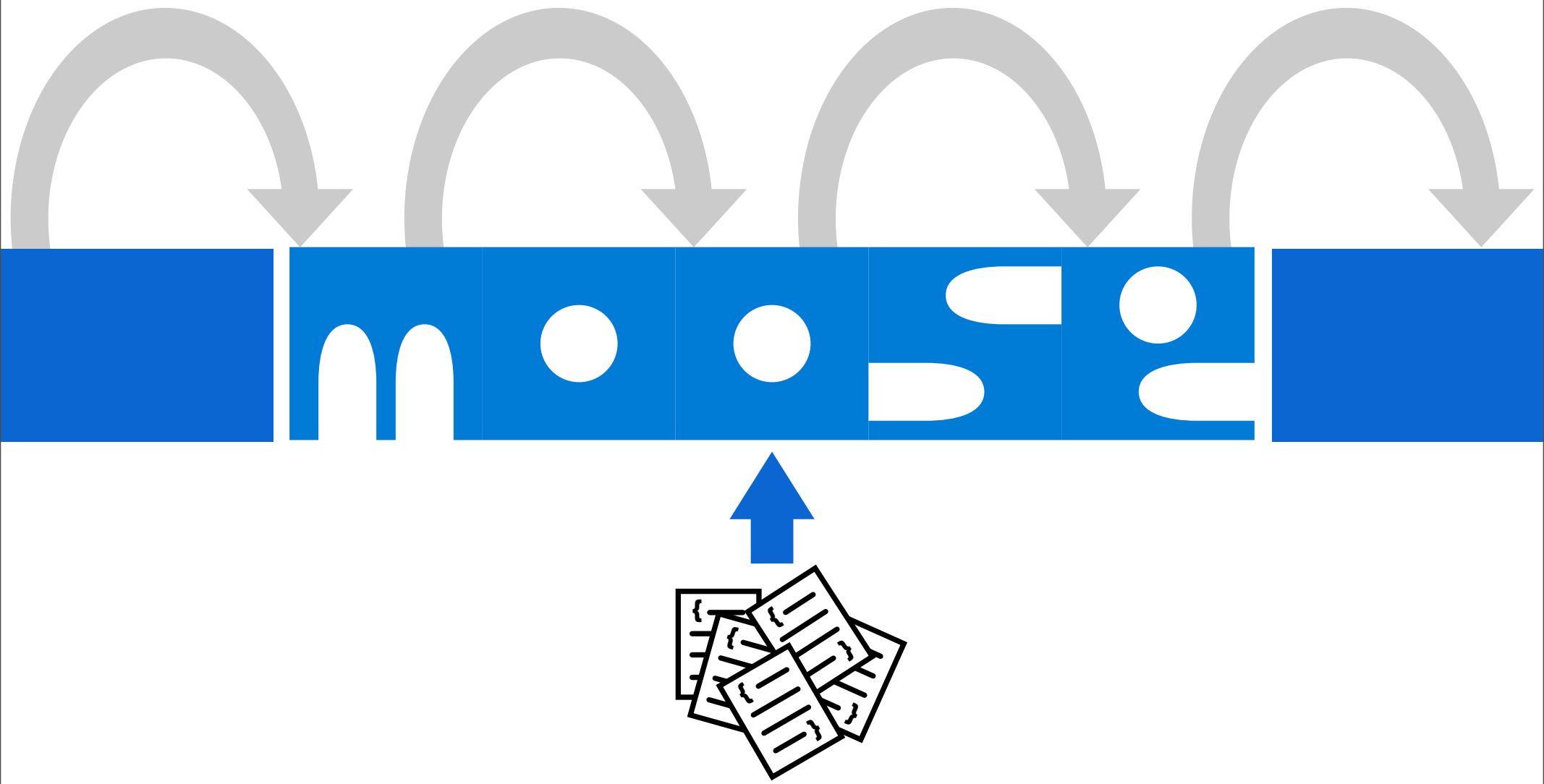


classes select: #isGod

McCabe = 21

LOC = 753,000

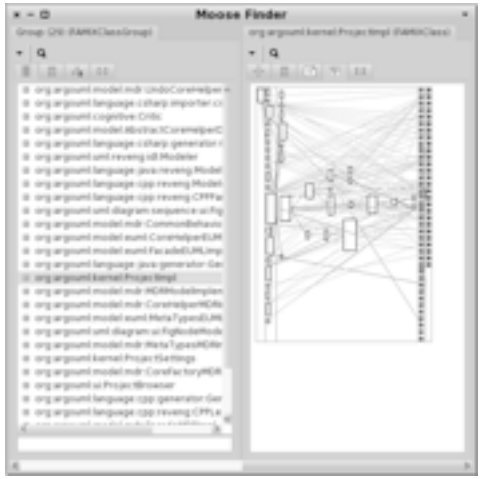


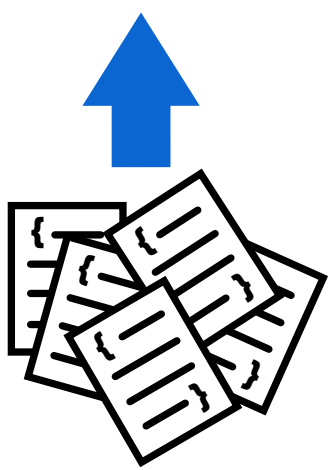


classes select: #isGod

McCabe = 21

LOC = 753,000

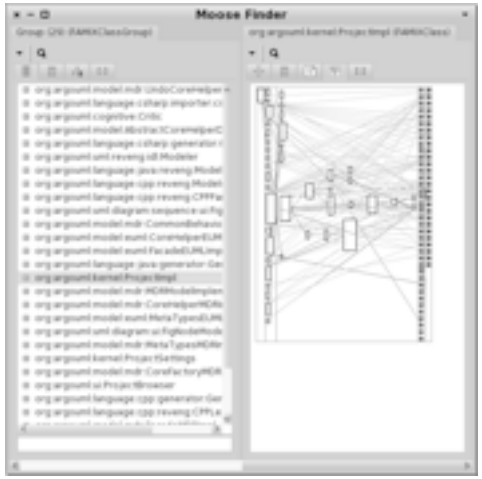


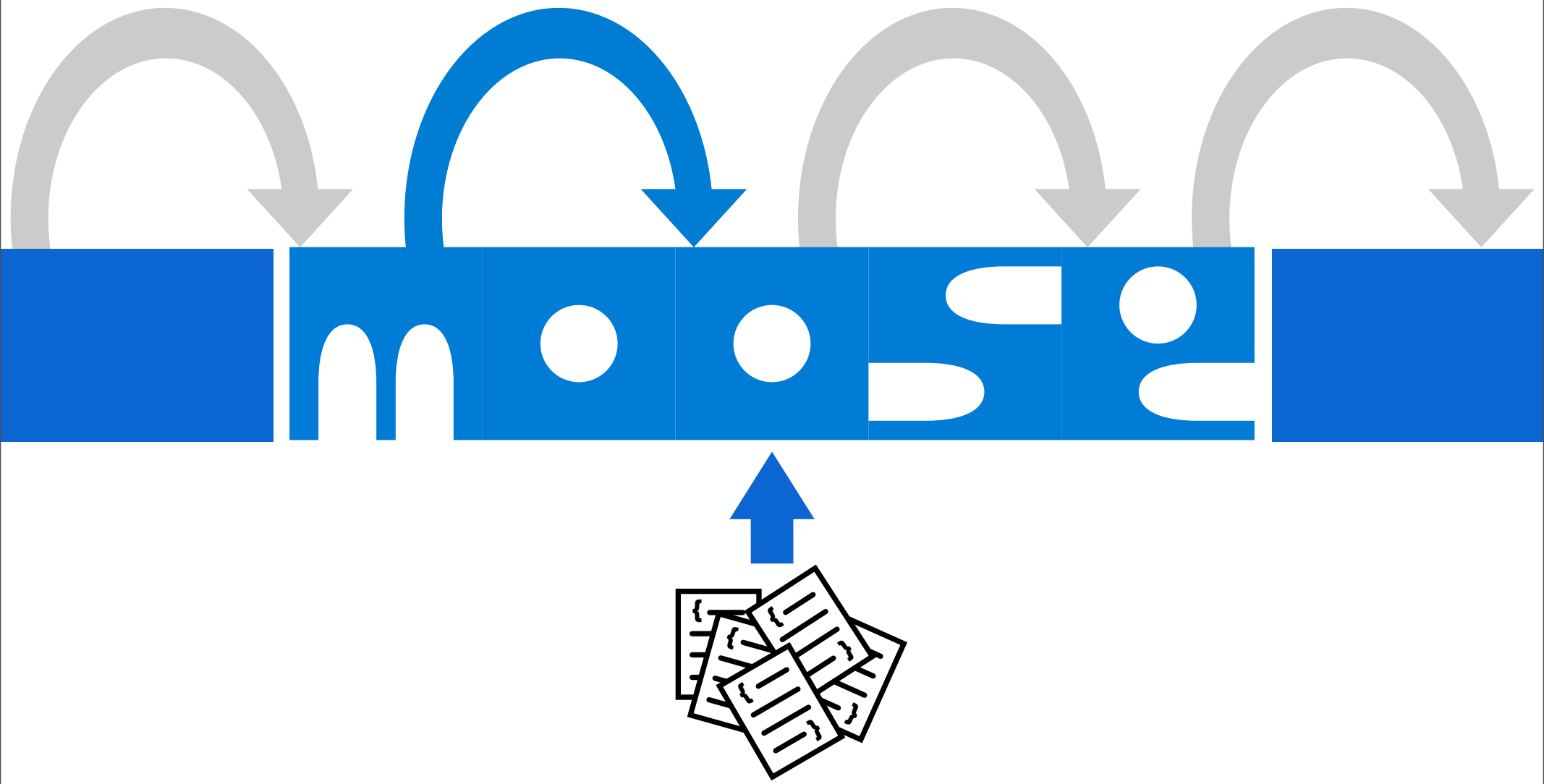


classes select: #isGod

McCabe = 21

LOC = 753,000

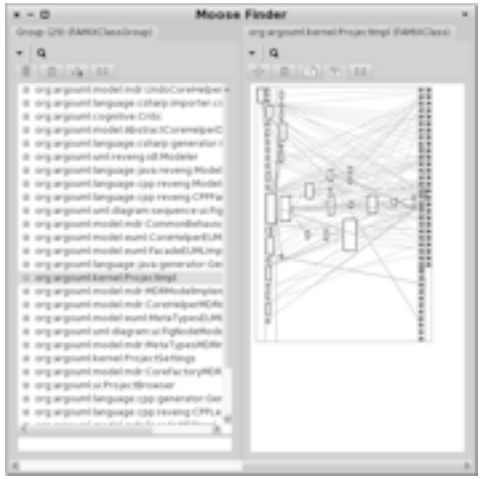


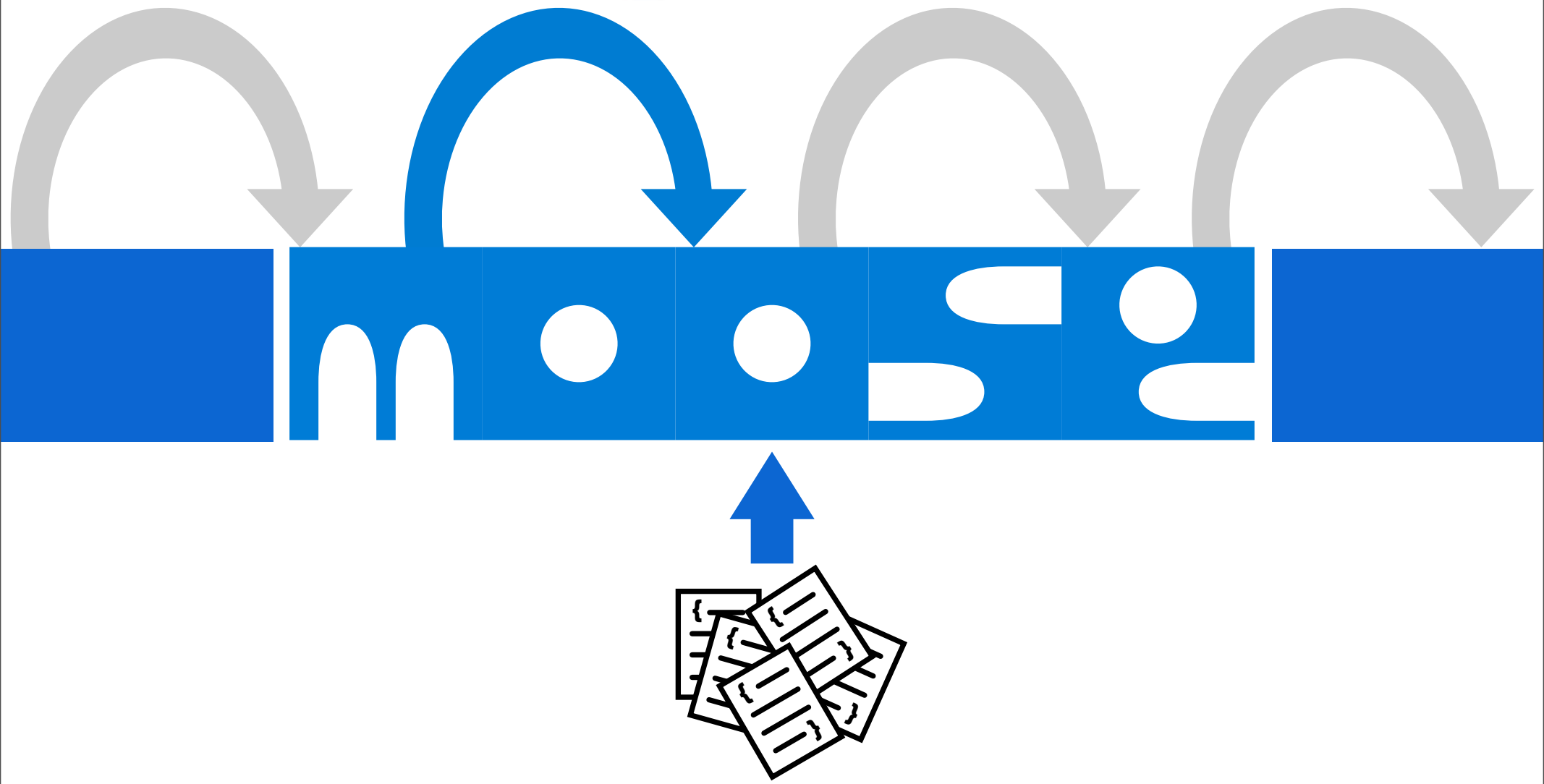


classes select: #isGod

McCabe = 21

LOC = 753,000

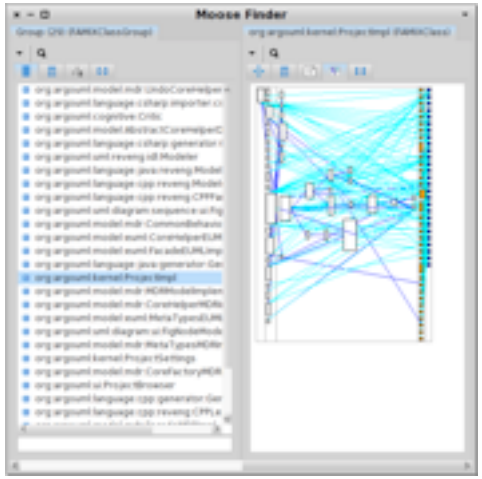


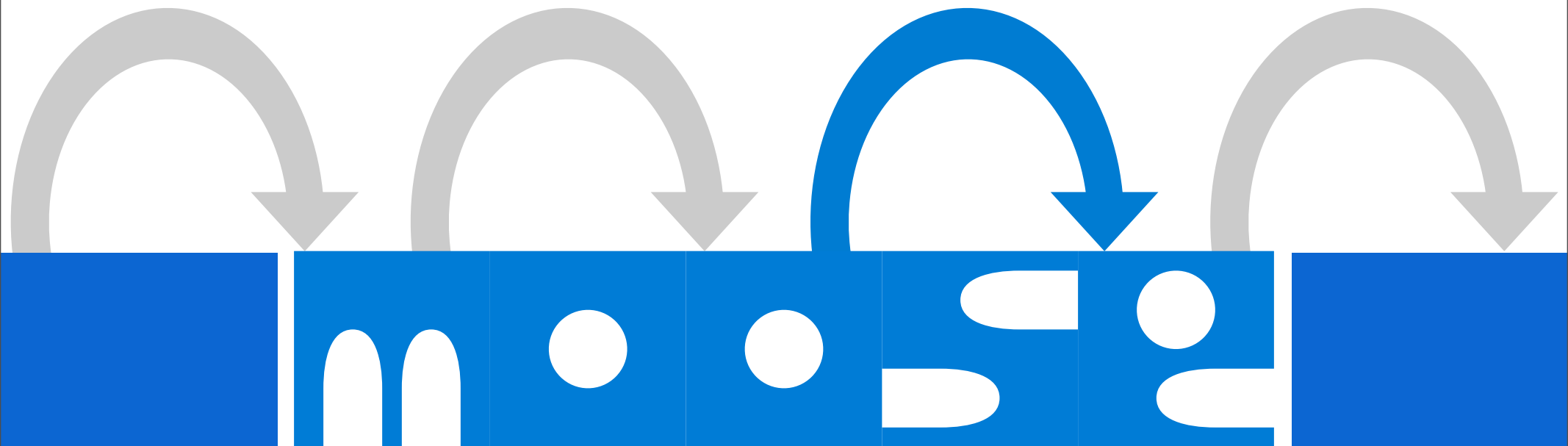


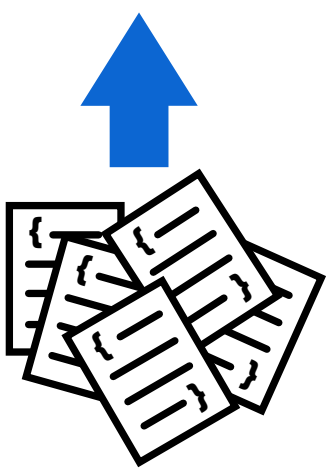
classes select: #isGod

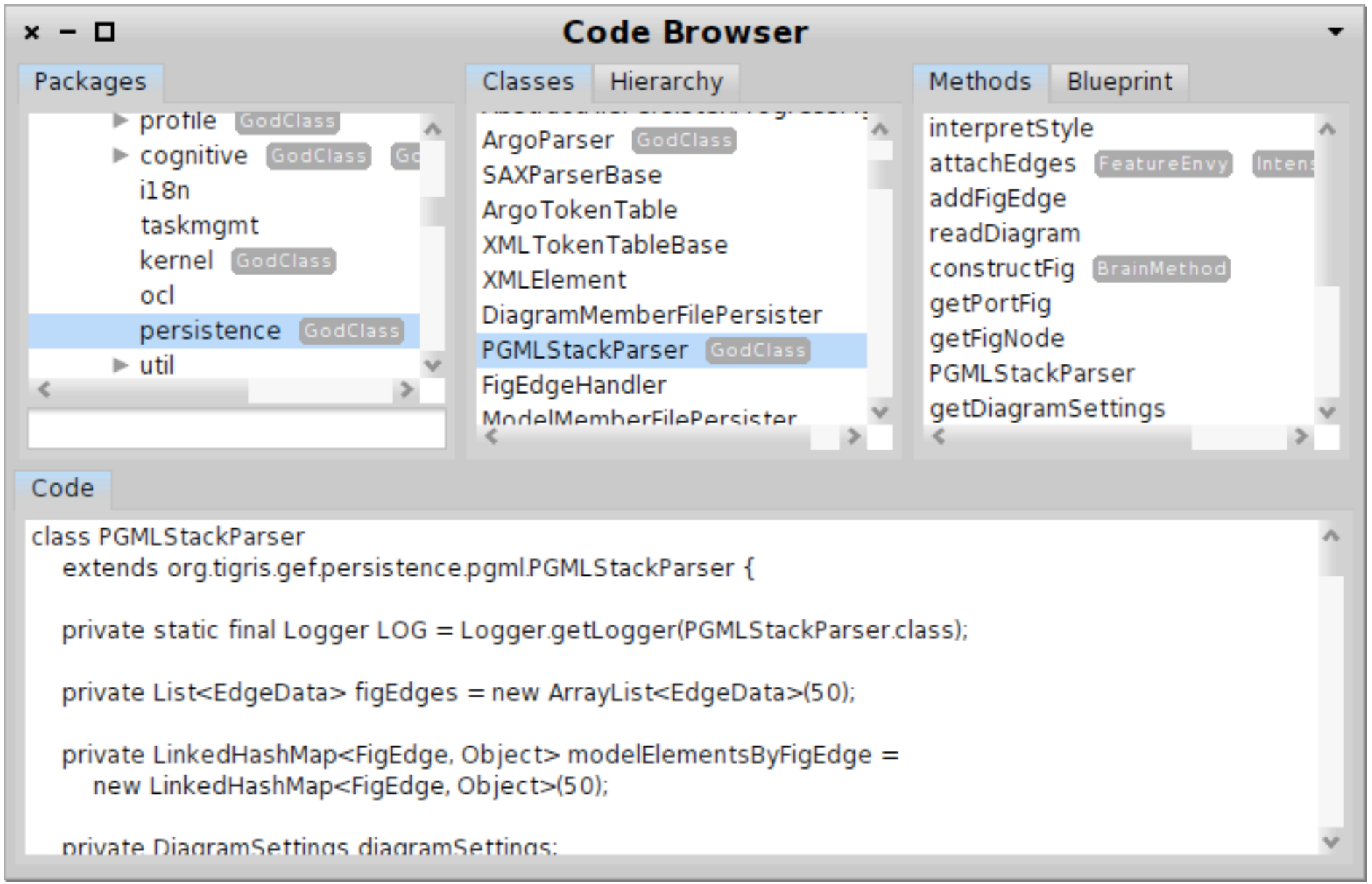
McCabe = 21

LOC = 753,000





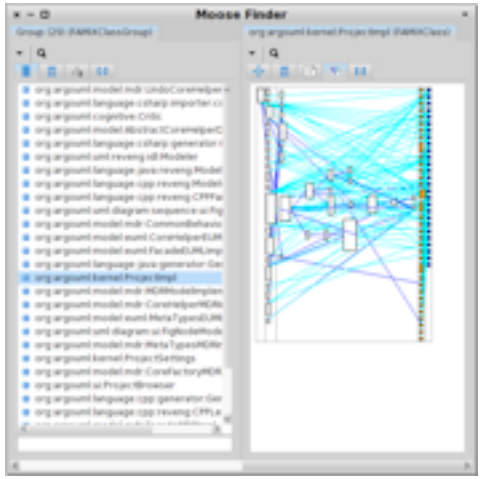


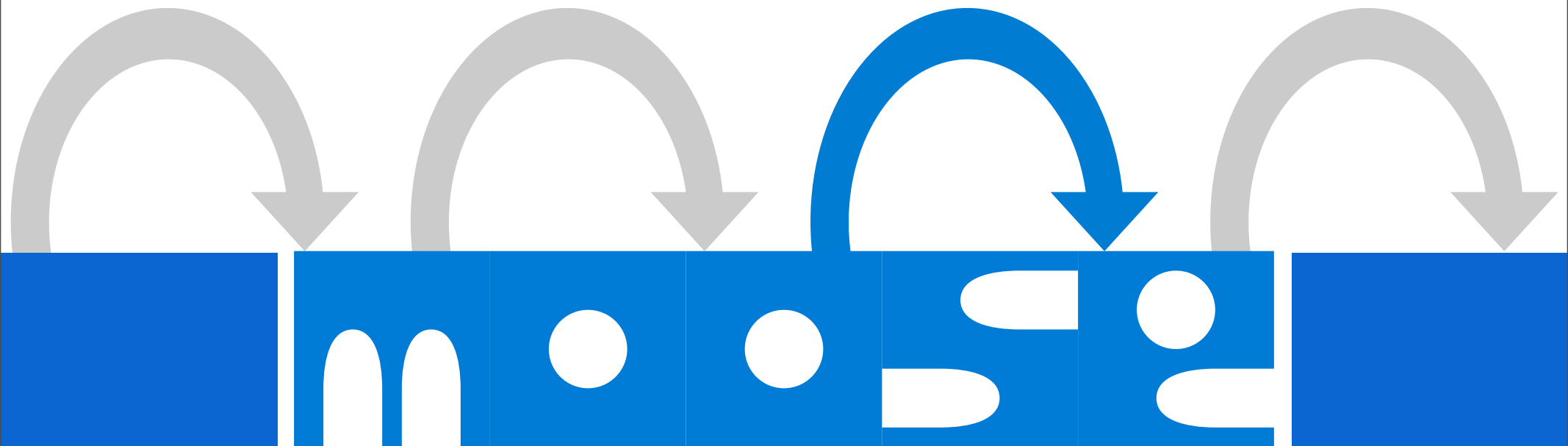

Code Browser
Packages
profile GodClass
cognitive GodClass
i18n
taskmgmt
kernel GodClass
ocl
persistence GodClass
util
Classes
Hierarchy
ArgoParser GodClass
SAXParserBase
ArgoTokenTable
XMLTokenTableBase
XMLElement
DiagramMemberFilePersister
PGMLStackParser GodClass
FigEdgeHandler
Methods
Blueprint
interpretStyle
attachEdges FeatureEnvy
addFigEdge
readDiagram
constructFig BrainMethod
getPortFig
getFigNode
PGMLStackParser
getDiagramSettings
Code
class PGMLStackParser
extends org.tigris.gef.persistence.pgml.PGMLStackParser {
private static final Logger LOG = Logger.getLogger(PGMLStackParser.class);
private List<EdgeData> figEdges = new ArrayList<EdgeData>(50);
private LinkedHashMap<FigEdge, Object> modelElementsByFigEdge =
new LinkedHashMap<FigEdge, Object>(50);
private DiagramSettings diagramSettings;

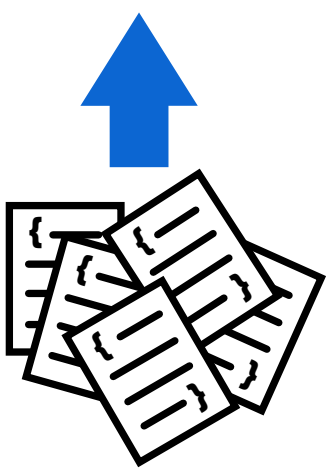
classes select: #isGod

McCabe = 21

LOC = 753,000



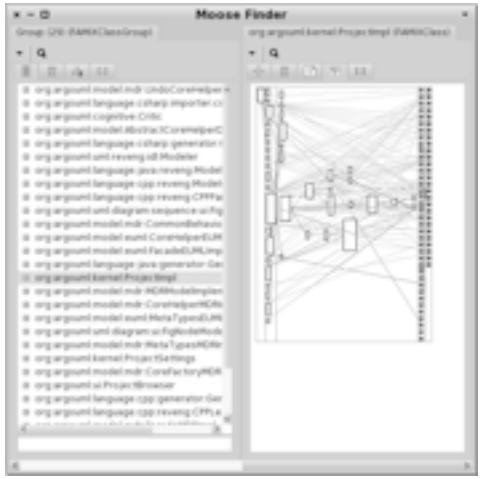


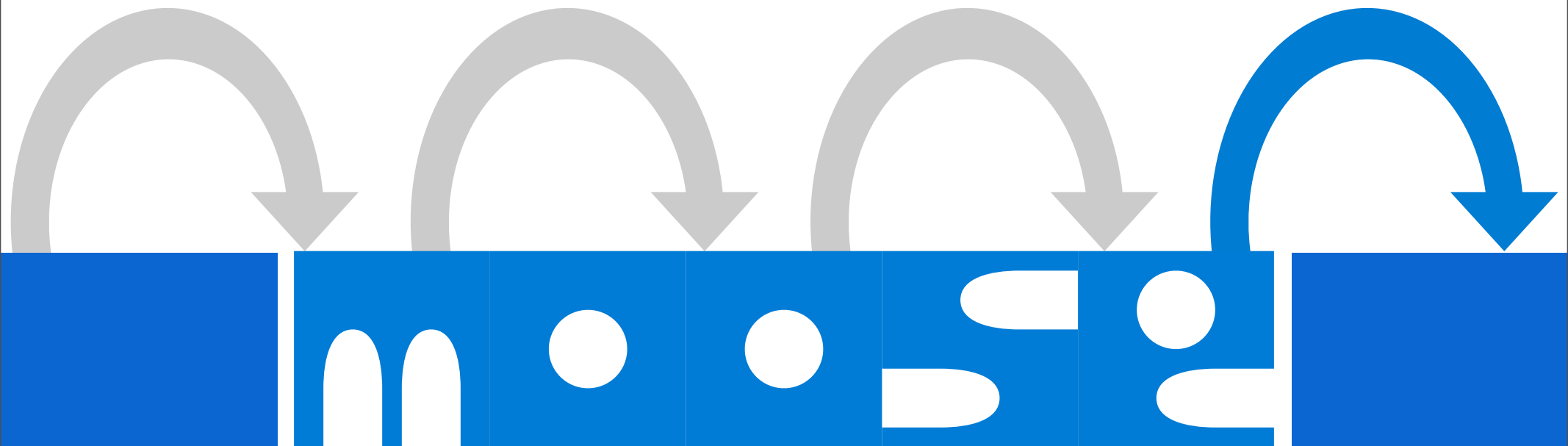


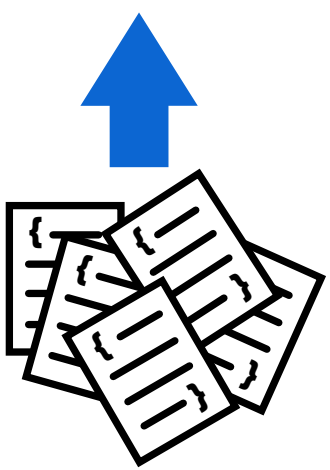
classes select: #isGod

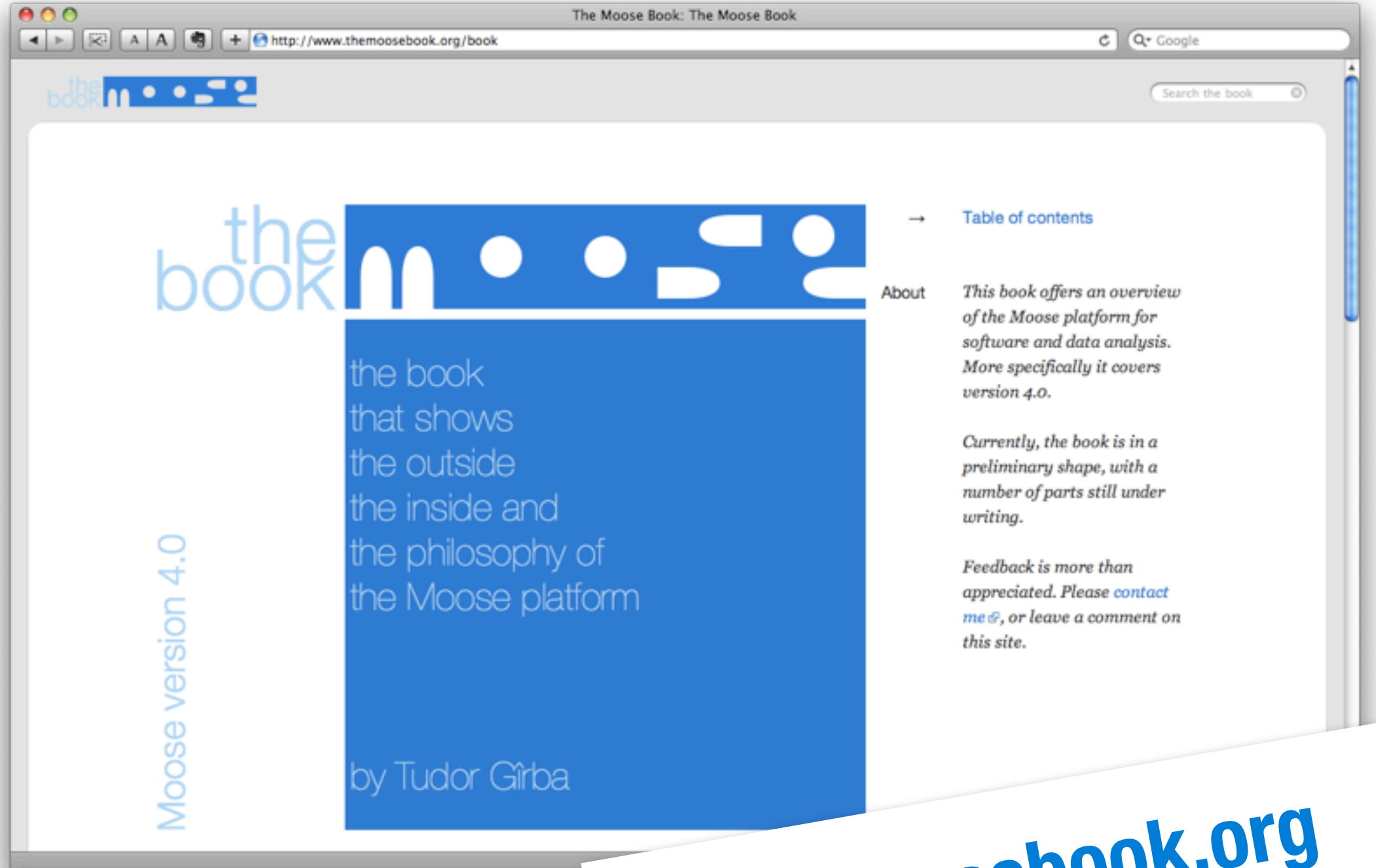
McCabe = 21

LOC = 753,000







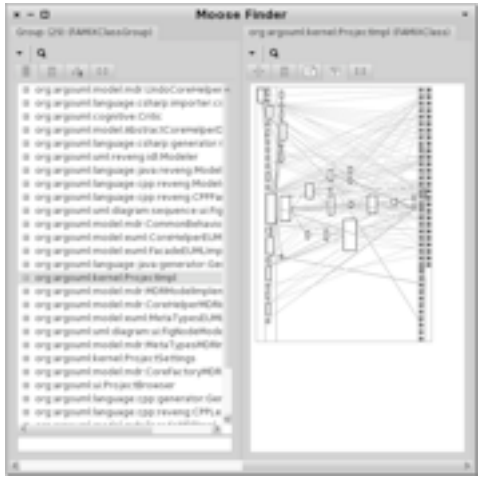


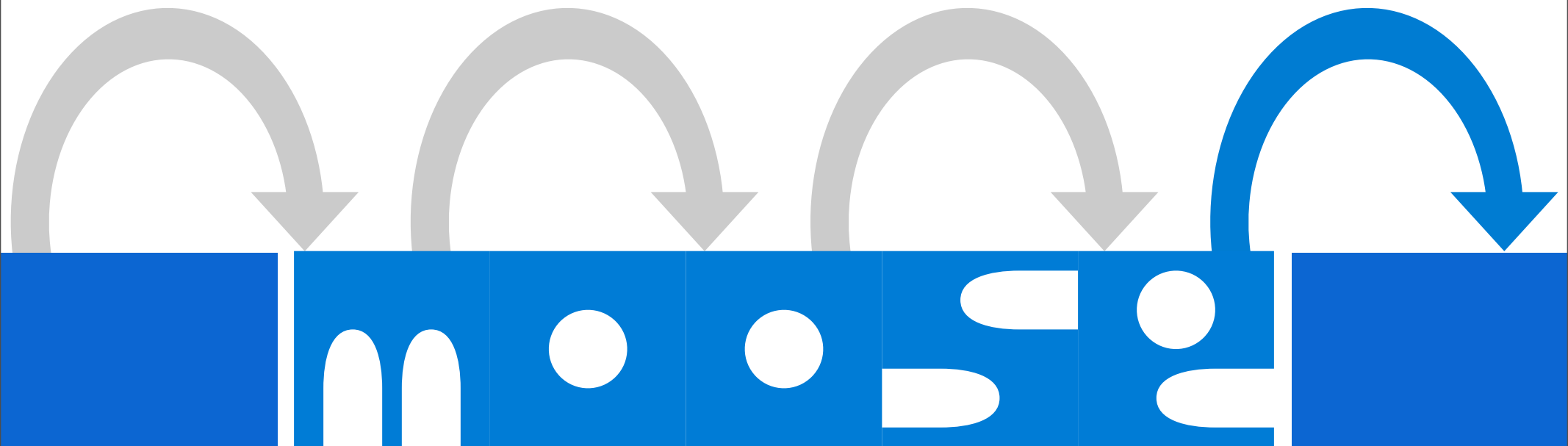
themoosebook.org

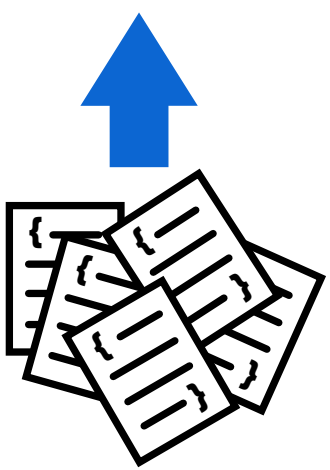
classes select: #isGod

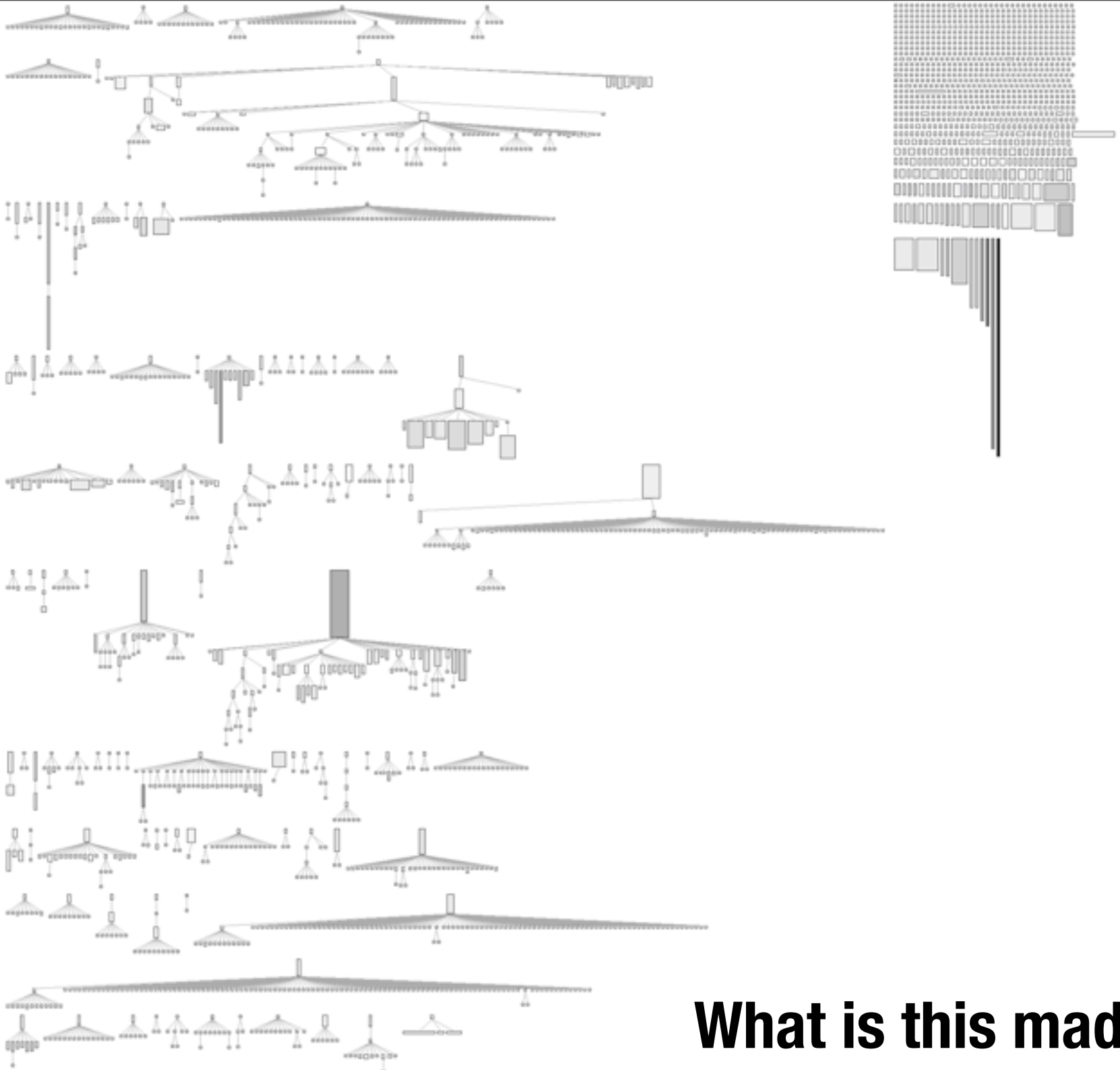
McCabe = 21

LOC = 753,000

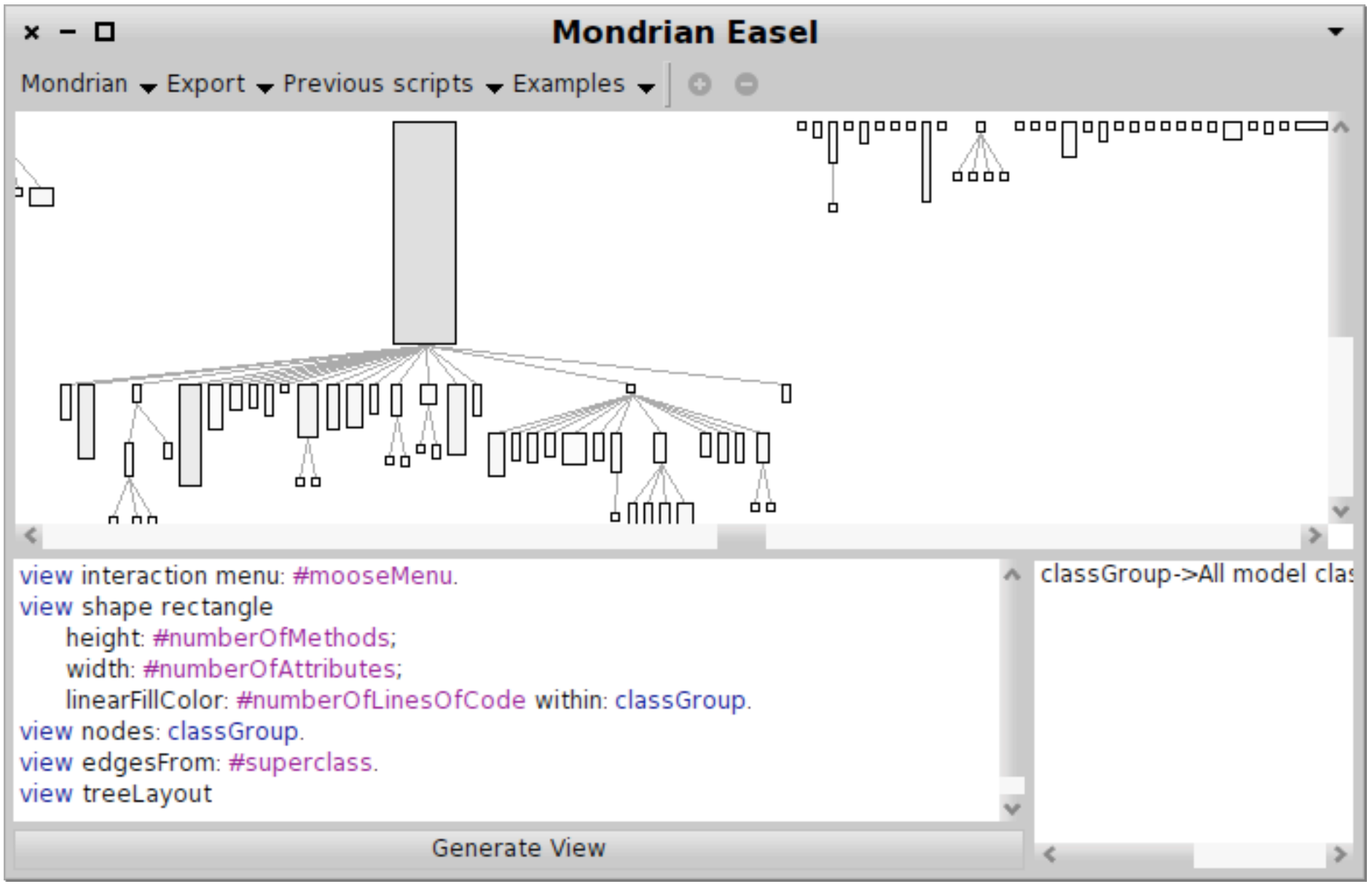


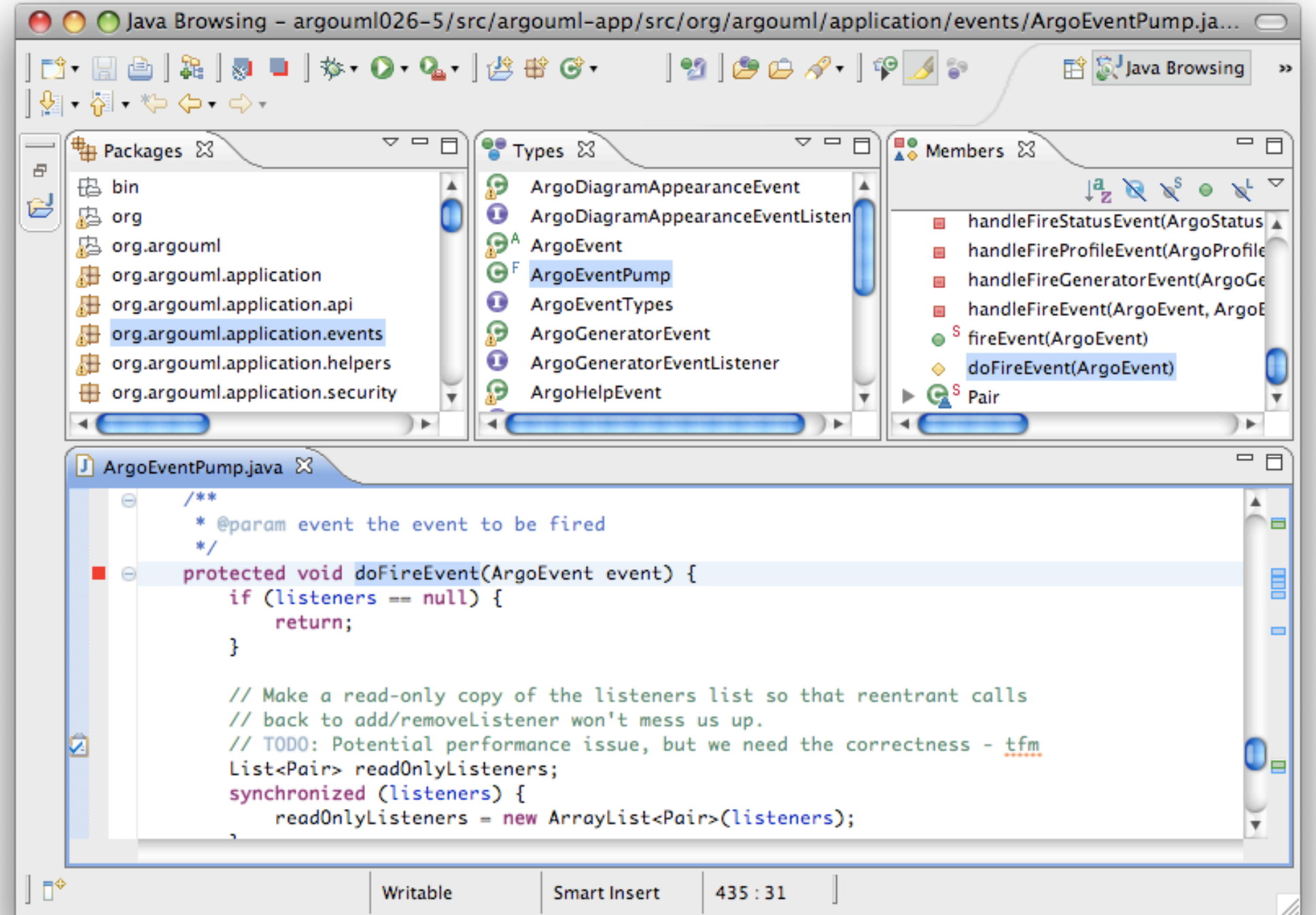




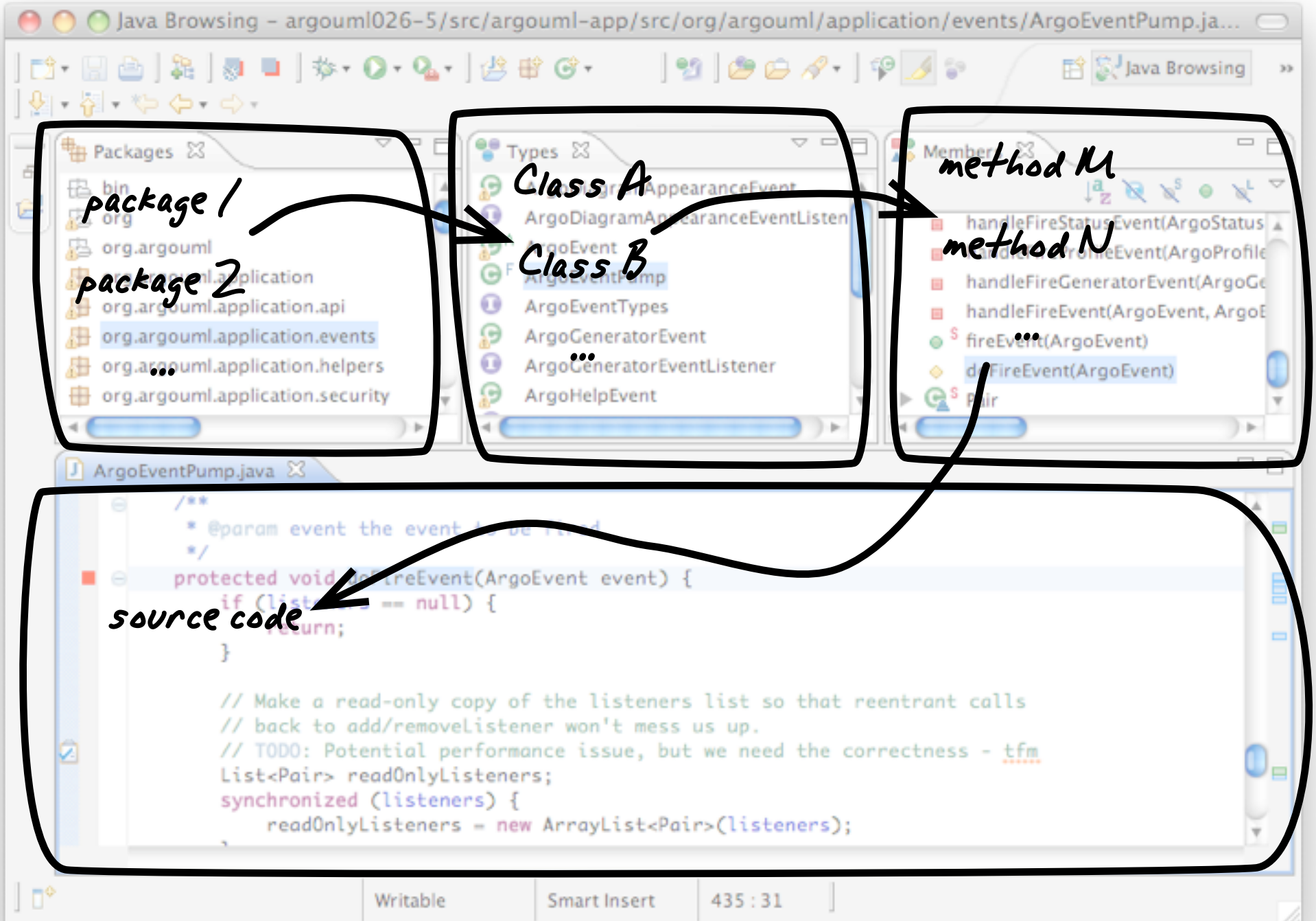


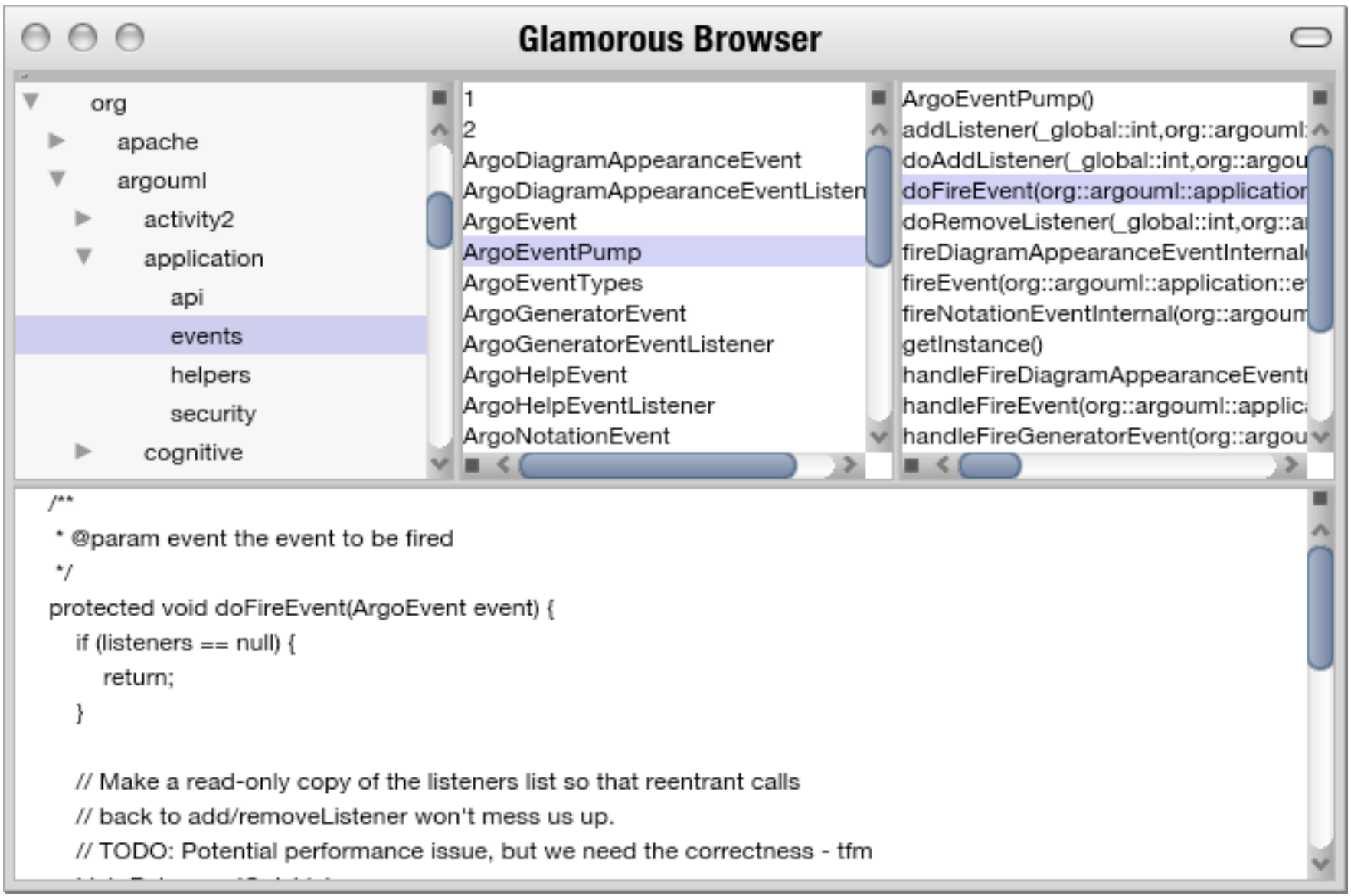
What is this made of?


Mondrian Easel
Mondrian
Export
Previous scripts
Examples
view interaction menu: #mooseMenu.
view shape rectangle
height: #numberOfMethods;
width: #numberOfAttributes;
linearFillColor: #numberOfLinesOfCode within: classGroup.
view nodes: classGroup.
view edgesFrom: #superclass.
view treeLayout
classGroup->All model cla
Generate View



What is this made of?

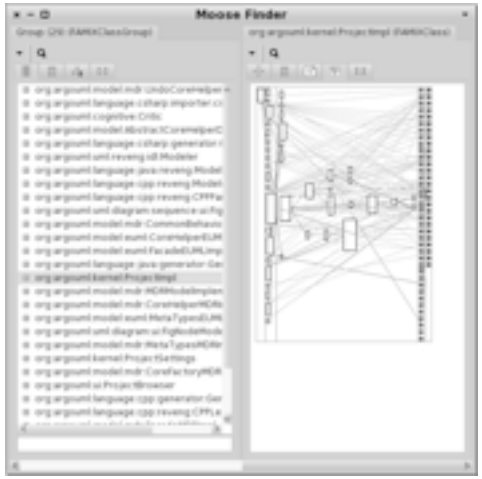

Java Browsing - argouml026-5/src/argouml-app/src/org/argouml/application/events/ArgoEventPump.ja...
Java Browsing
Packages
bin
org
org.argouml
org.argouml.application
org.argouml.application.api
org.argouml.application.events
org.argouml.application.helpers
org.argouml.application.security
...
package 1
package 2
Types
ArgoDiagramAppearanceEventListen
ArgoEvent
ArgoEventPump
ArgoEventTypes
ArgoGeneratorEvent
ArgoGeneratorEventListener
ArgoHelpEvent
...
Class A
Class B
Member
handleFireStatusEvent(ArgoStatus
handleFireGeneratorEvent(ArgoGe
handleFireEvent(ArgoEvent, ArgoE
fireEvent(ArgoEvent)
doFireEvent(ArgoEvent)
...
method M
method N
ArgoEventPump.java
protected void doFireEvent(ArgoEvent event) {
if (listeners == null) {
return;
}
// Make a read-only copy of the listeners list so that reentrant calls
// back to add/removeListener won't mess us up.
// TODO: Potential performance issue, but we need the correctness - tfm
List<Pair> readOnlyListeners;
synchronized (listeners) {
readOnlyListeners = new ArrayList<Pair>(listeners);
source code
Writable
Smart Insert
435 : 31

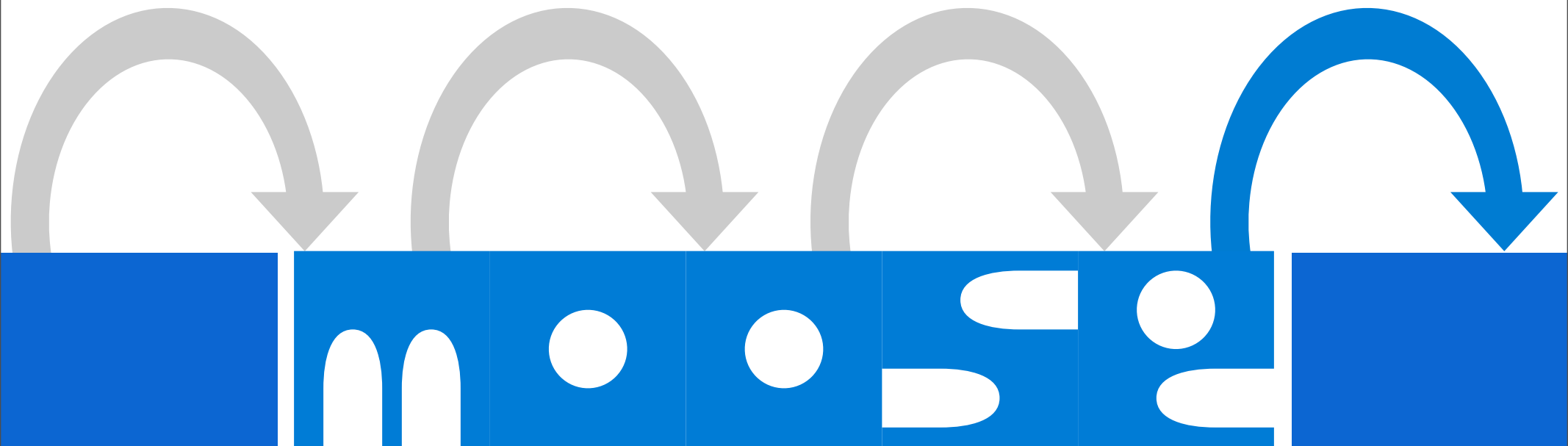

Glamorous Browser
org
apache
argouml
activity2
application
api
events
helpers
security
cognitive
1
2
ArgoDiagramAppearanceEvent
ArgoDiagramAppearanceEventListen
ArgoEvent
ArgoEventPump
ArgoEventTypes
ArgoGeneratorEvent
ArgoGeneratorEventListener
ArgoHelpEvent
ArgoHelpEventListener
ArgoNotationEvent
ArgoEventPump()
addListener(_global::int,org::argouml:
doAddListener(_global::int,org::argou
doFireEvent(org::argouml::applicatior
doRemoveListener(_global::int,org::a
fireDiagramAppearanceEventInternal
fireEvent(org::argouml::application::e
fireNotationEventInternal(org::argoum
getInstance()
handleFireDiagramAppearanceEvent
handleFireEvent(org::argouml::applic
handleFireGeneratorEvent(org::argou
/**
* @param event the event to be fired
*/
protected void doFireEvent(ArgoEvent event) {
if (listeners == null) {
return;
}
// Make a read-only copy of the listeners list so that reentrant calls
// back to add/removeListener won't mess us up.
// TODO: Potential performance issue, but we need the correctness - tfm

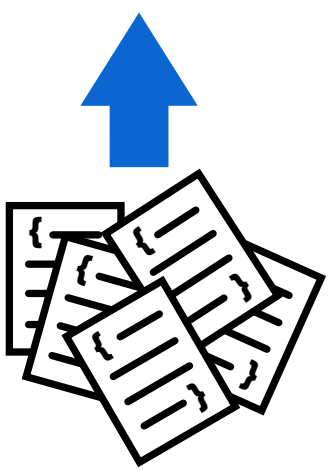
classes select: #isGod

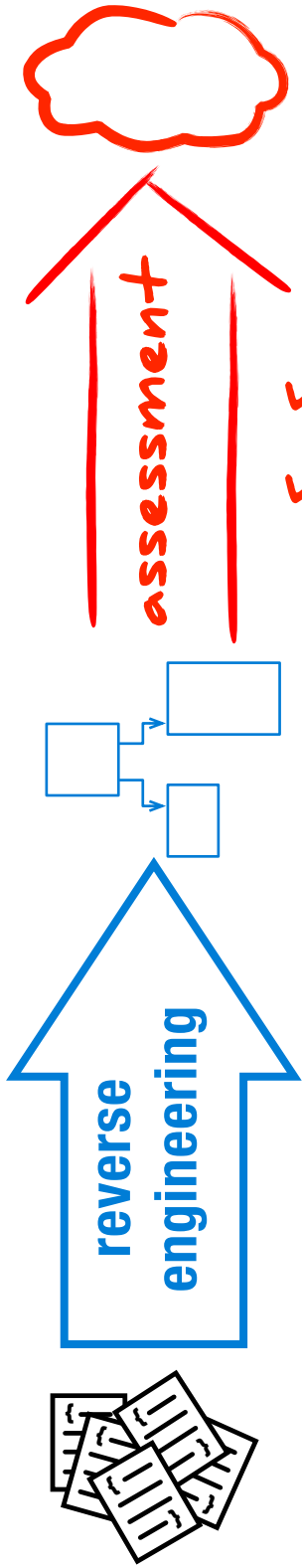
McCabe = 21

LOC = 753,000









what is the current situation?
what can we do about it?

Handling Moose

5 days to get up to speed

21 days to get proficient

Software engineers that extended Moose

Delphi/Java programmers

- customized models
- extended meta models
- defined specific rules

Smalltalk programmers

- new tools to support code integration

Moose supports Your definition of quality/risk

Example

(1) Getting bug maps

(2) What info to collect?

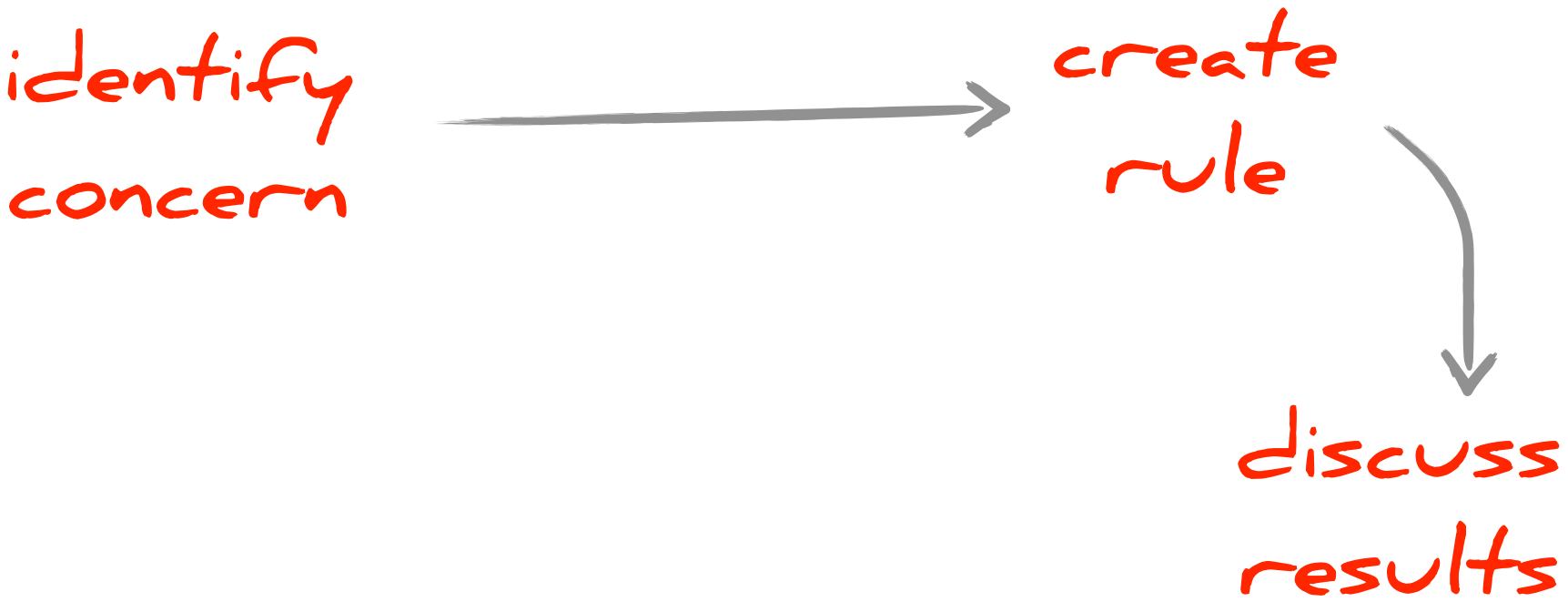
(3) How to present it?

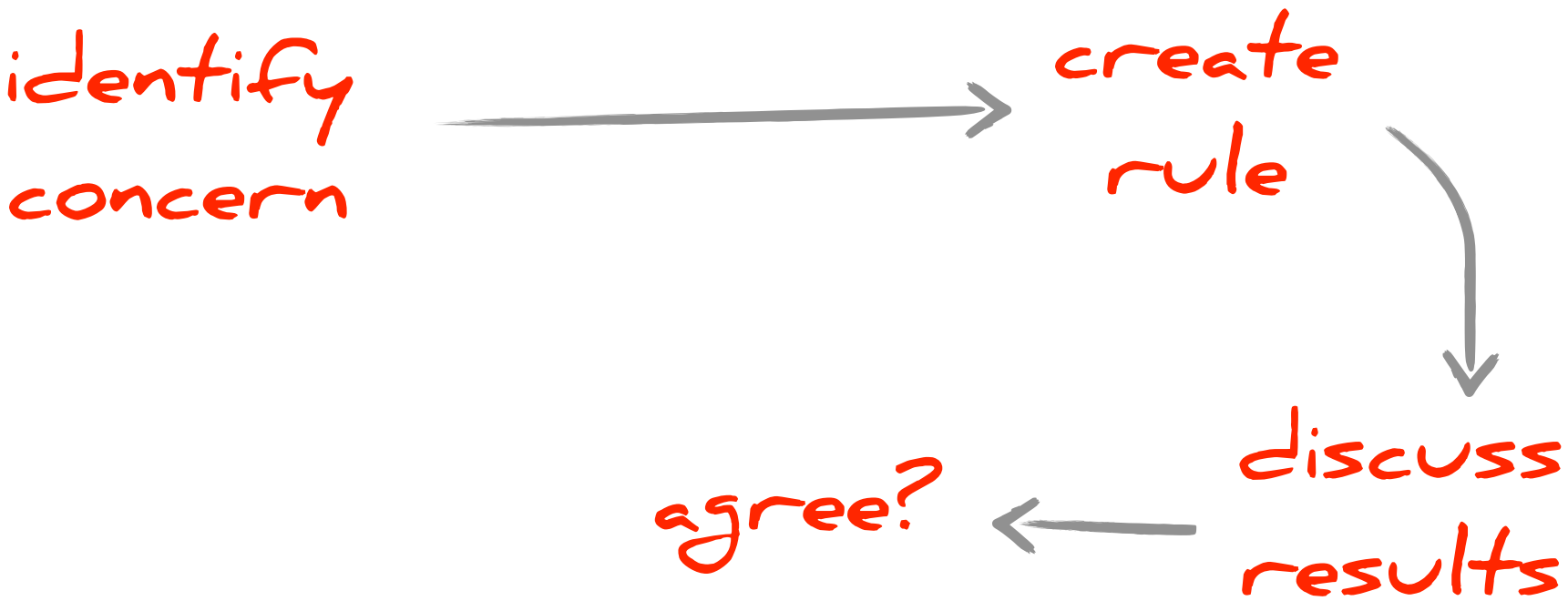
Moose gives you the opportunity to control your process.

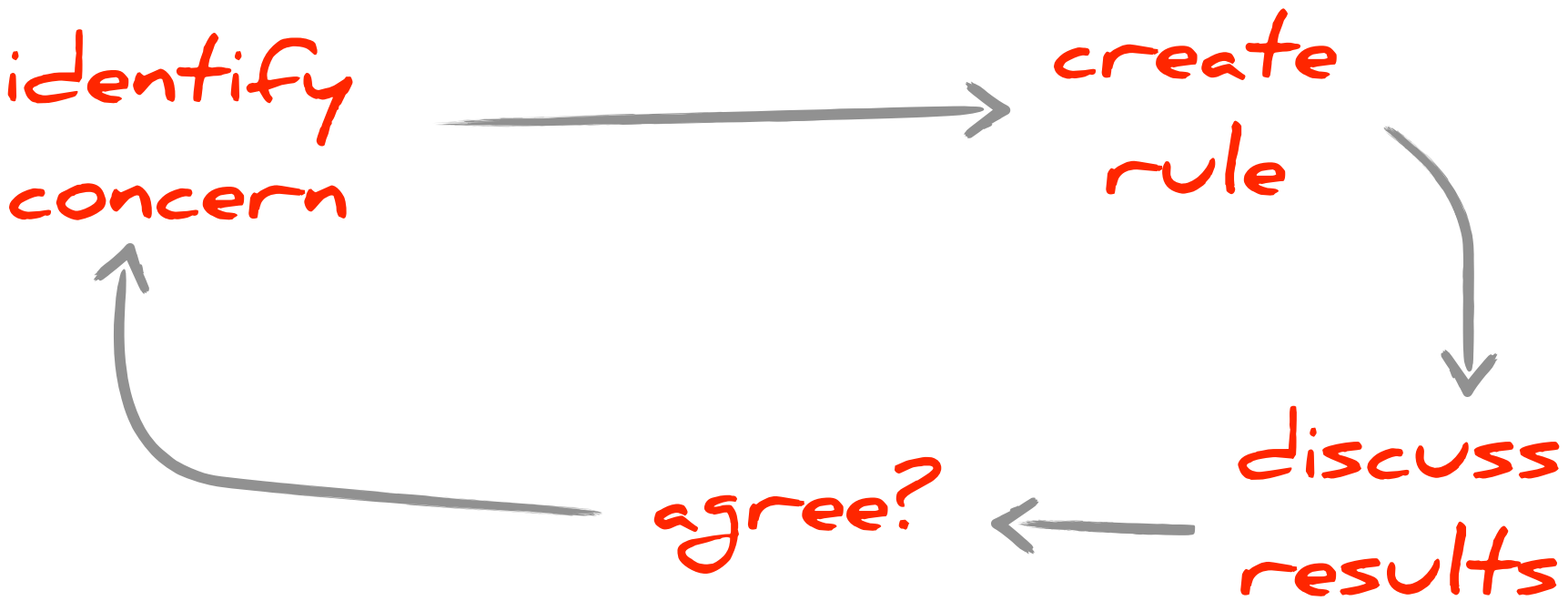
identify
concern

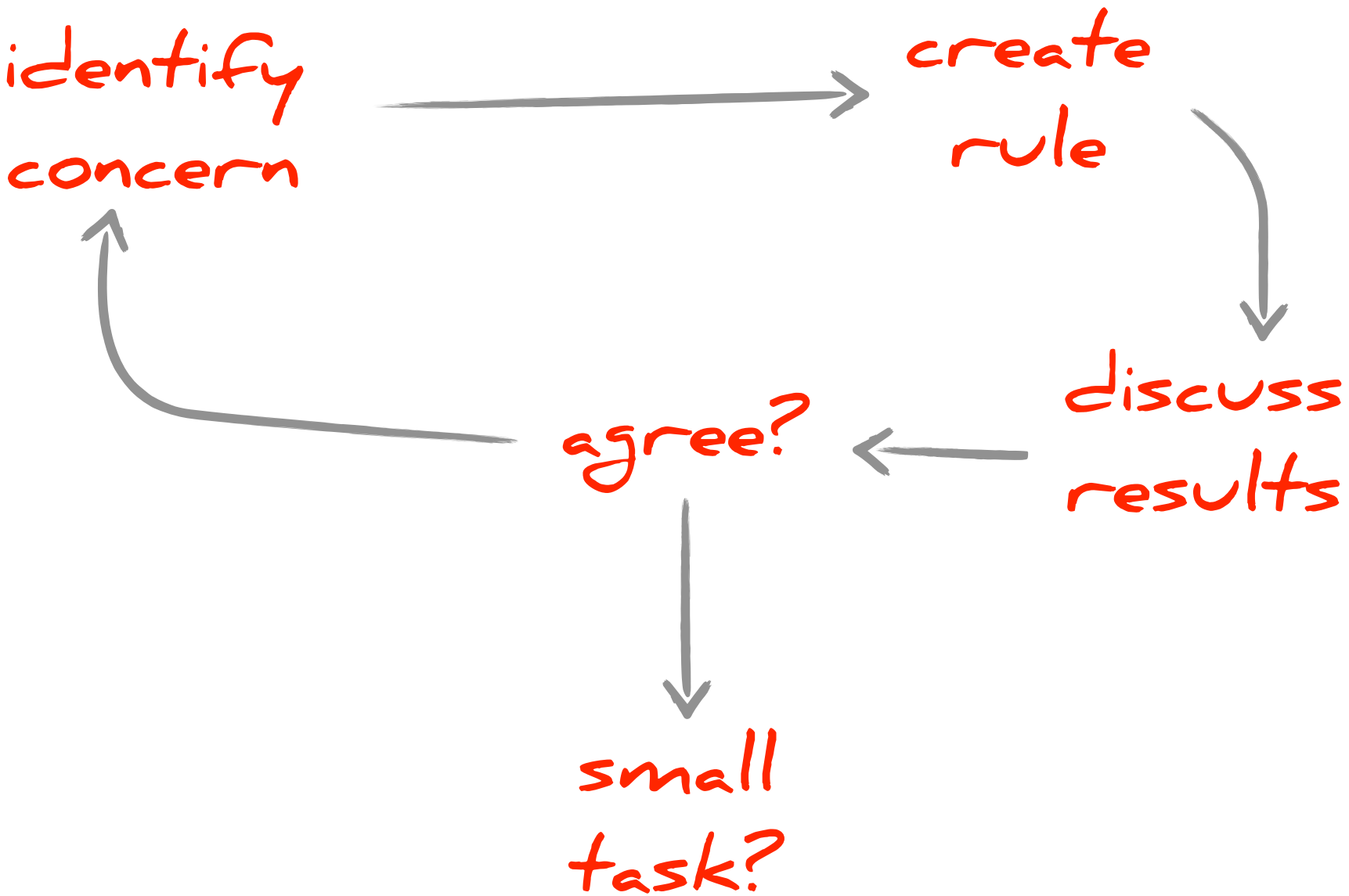
identify concern

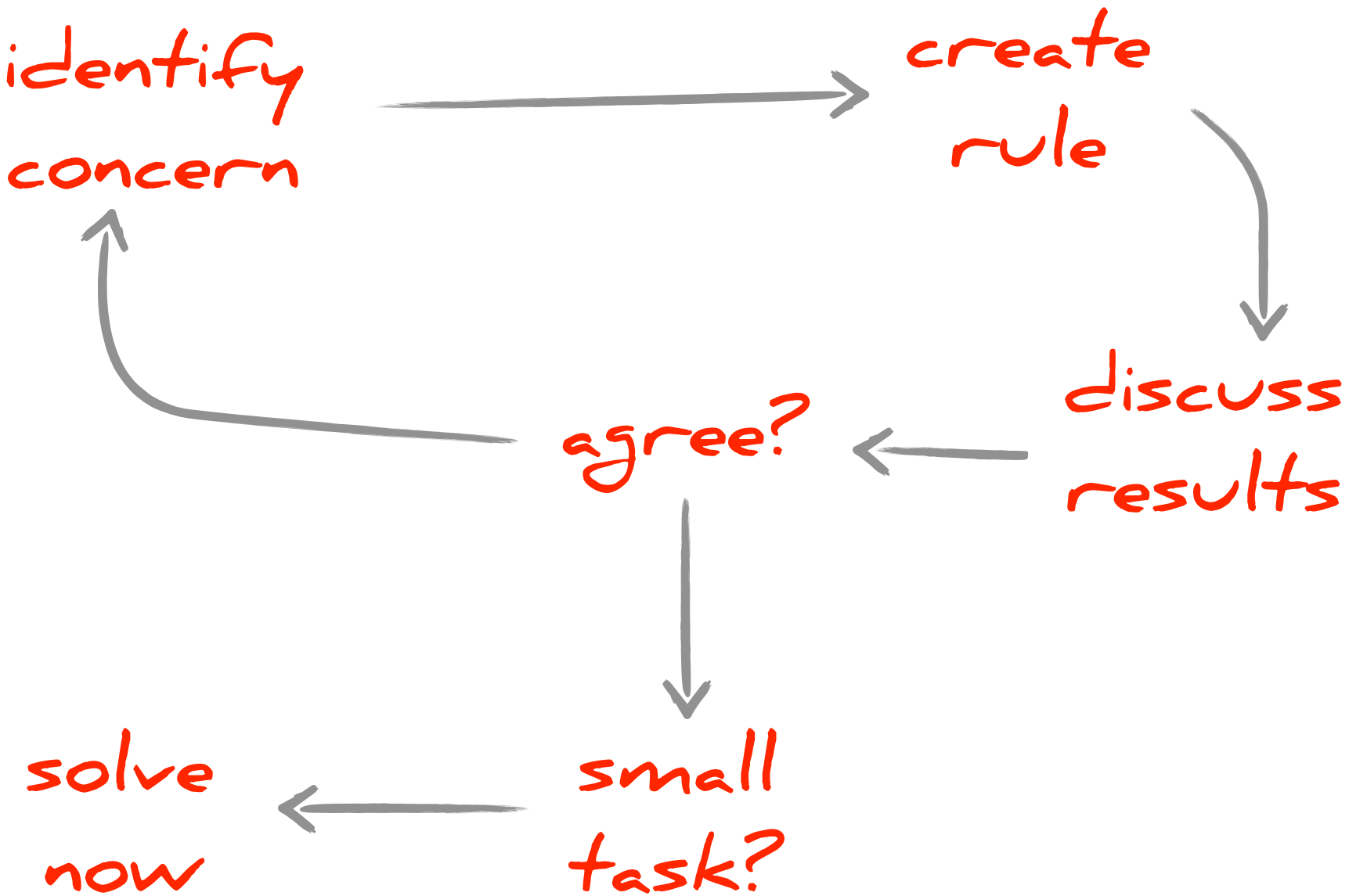
create rule

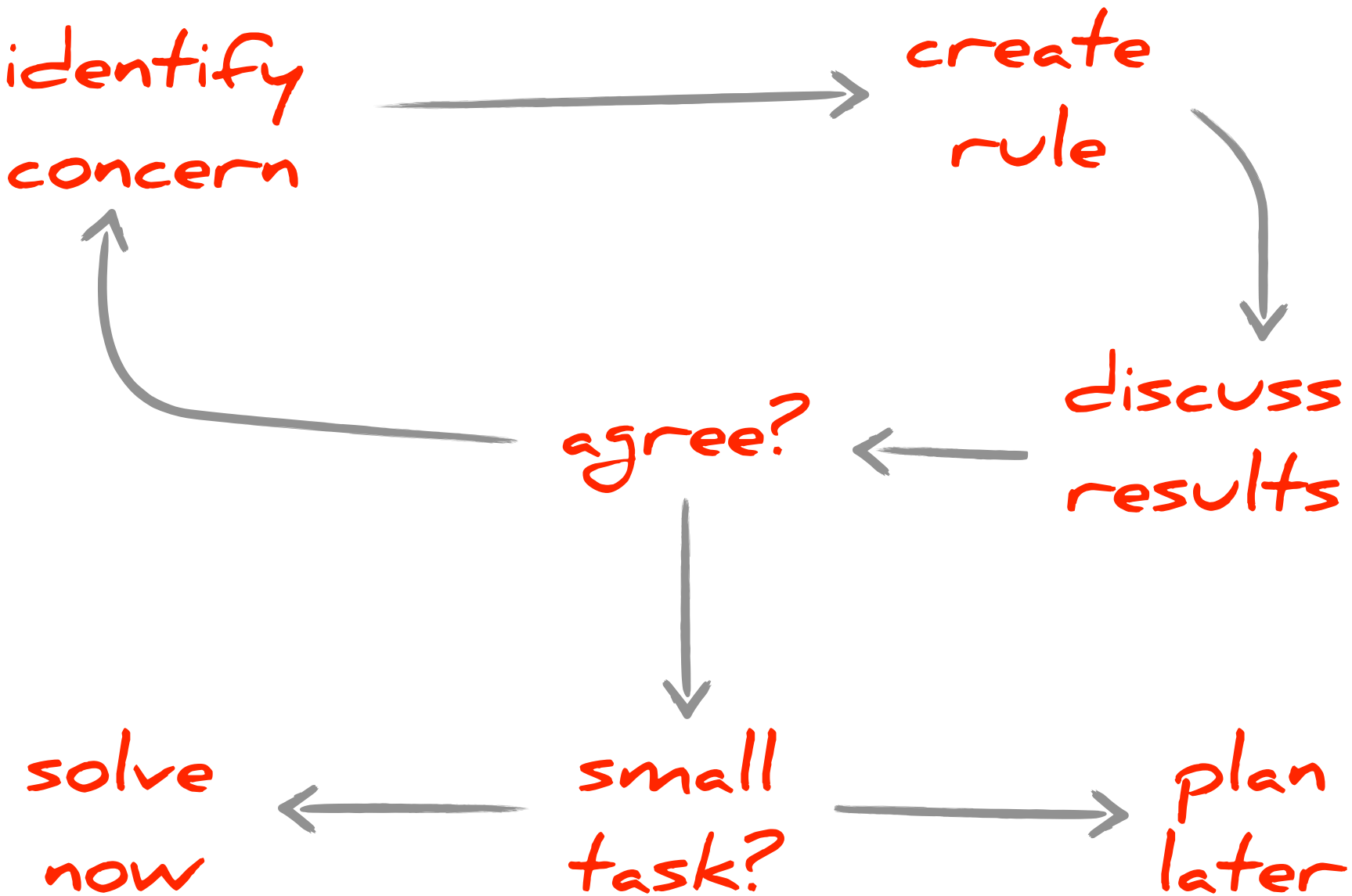

identify
concern
create
rule
discuss
results


identify concern
create rule
discuss results
agree?

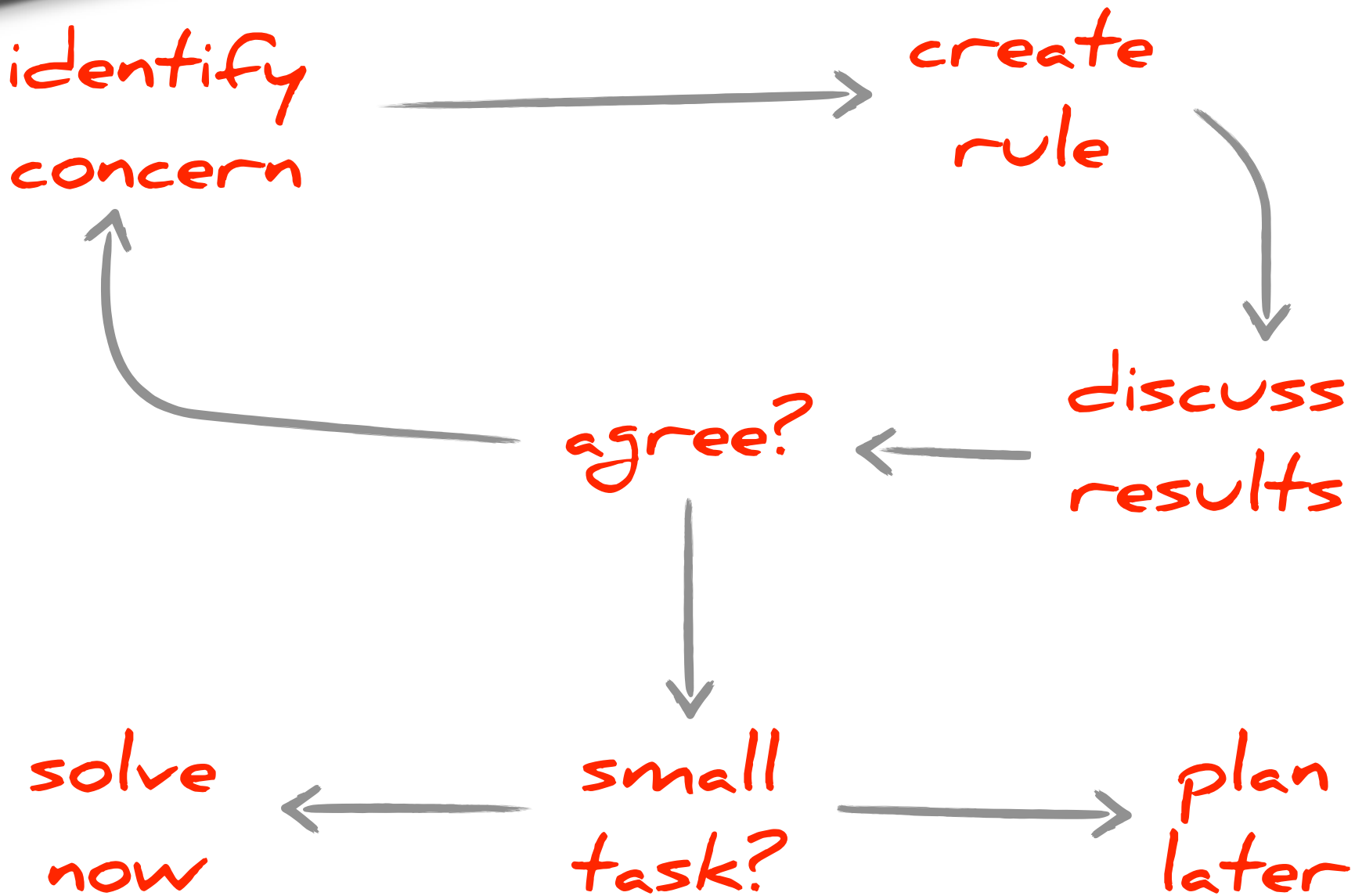

identify concern
create rule
discuss results
agree?

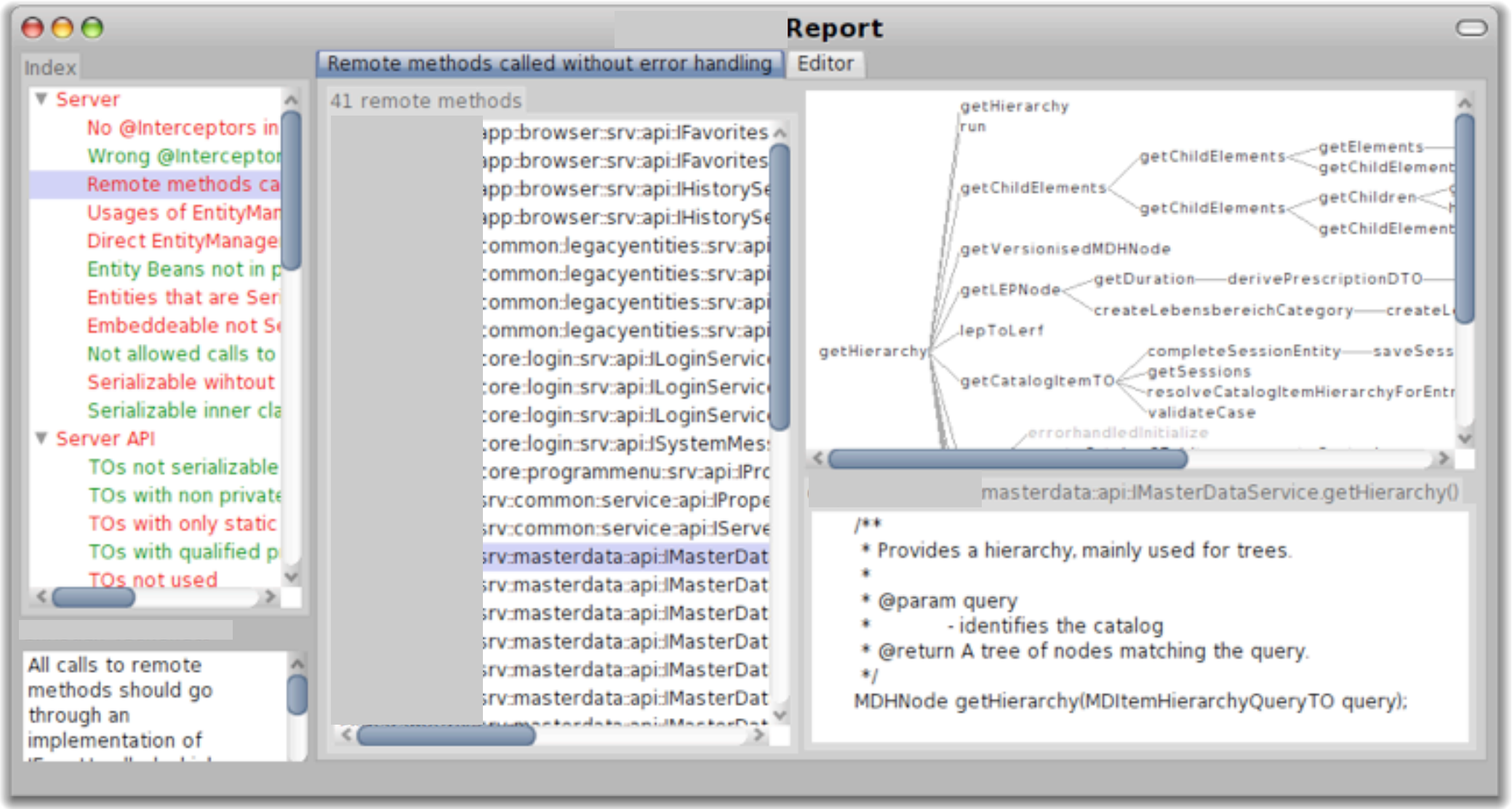

identify concern
create rule
discuss results
agree?
small task?

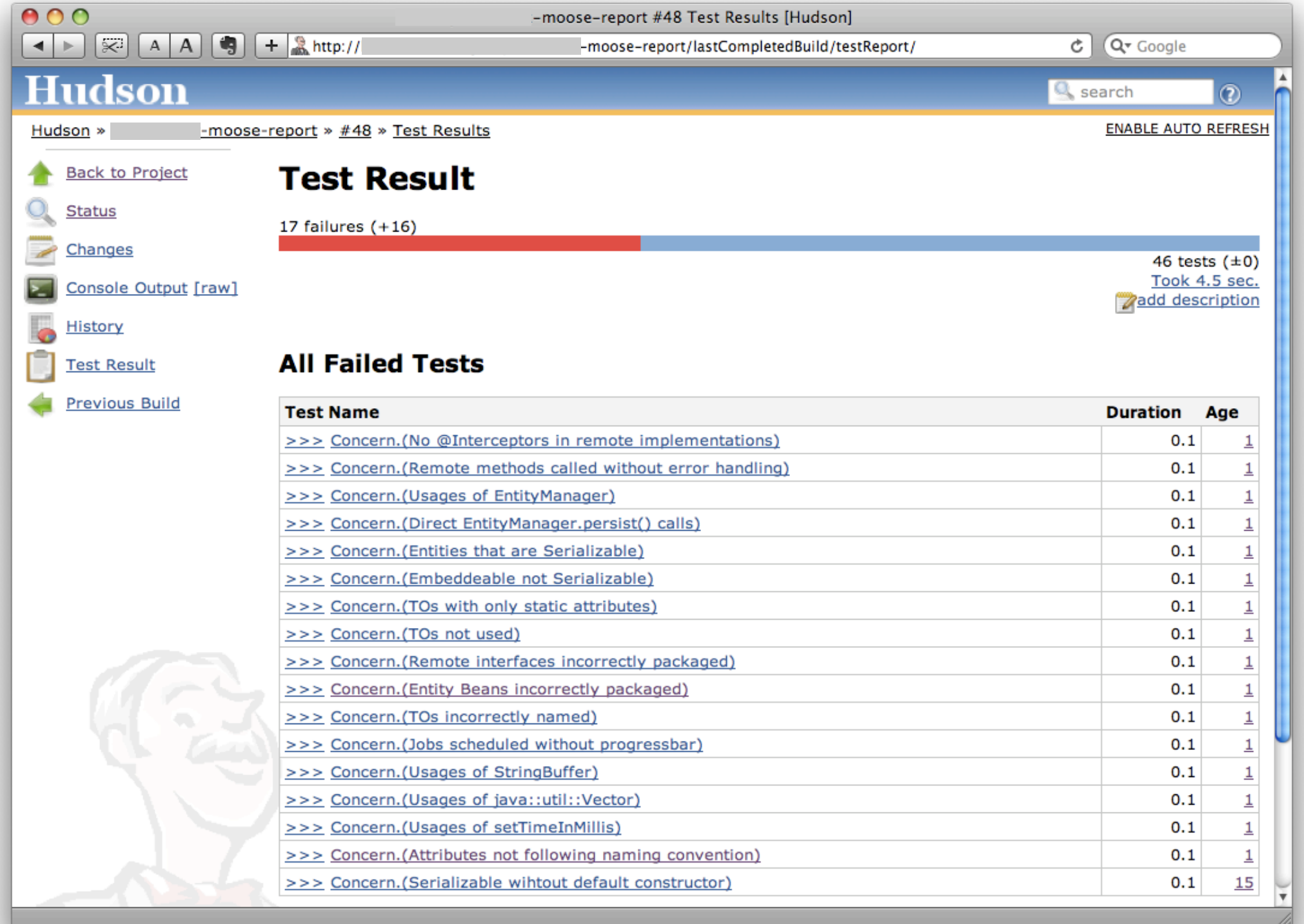

identify concern
create rule
discuss results
agree?
small task?
solve now

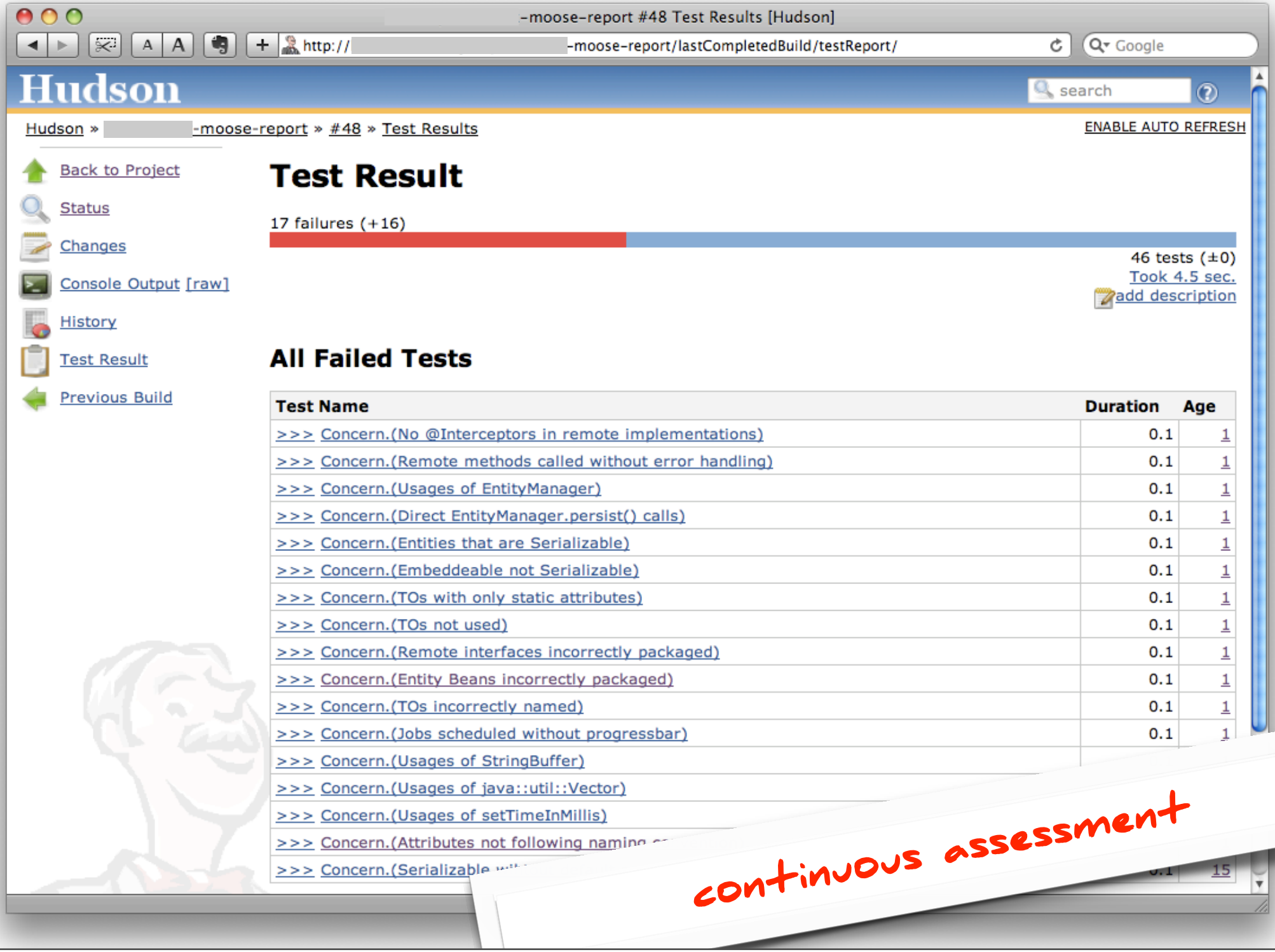

identify concern
create rule
discuss results
agree?
small task?
solve now
plan later

daily assessment

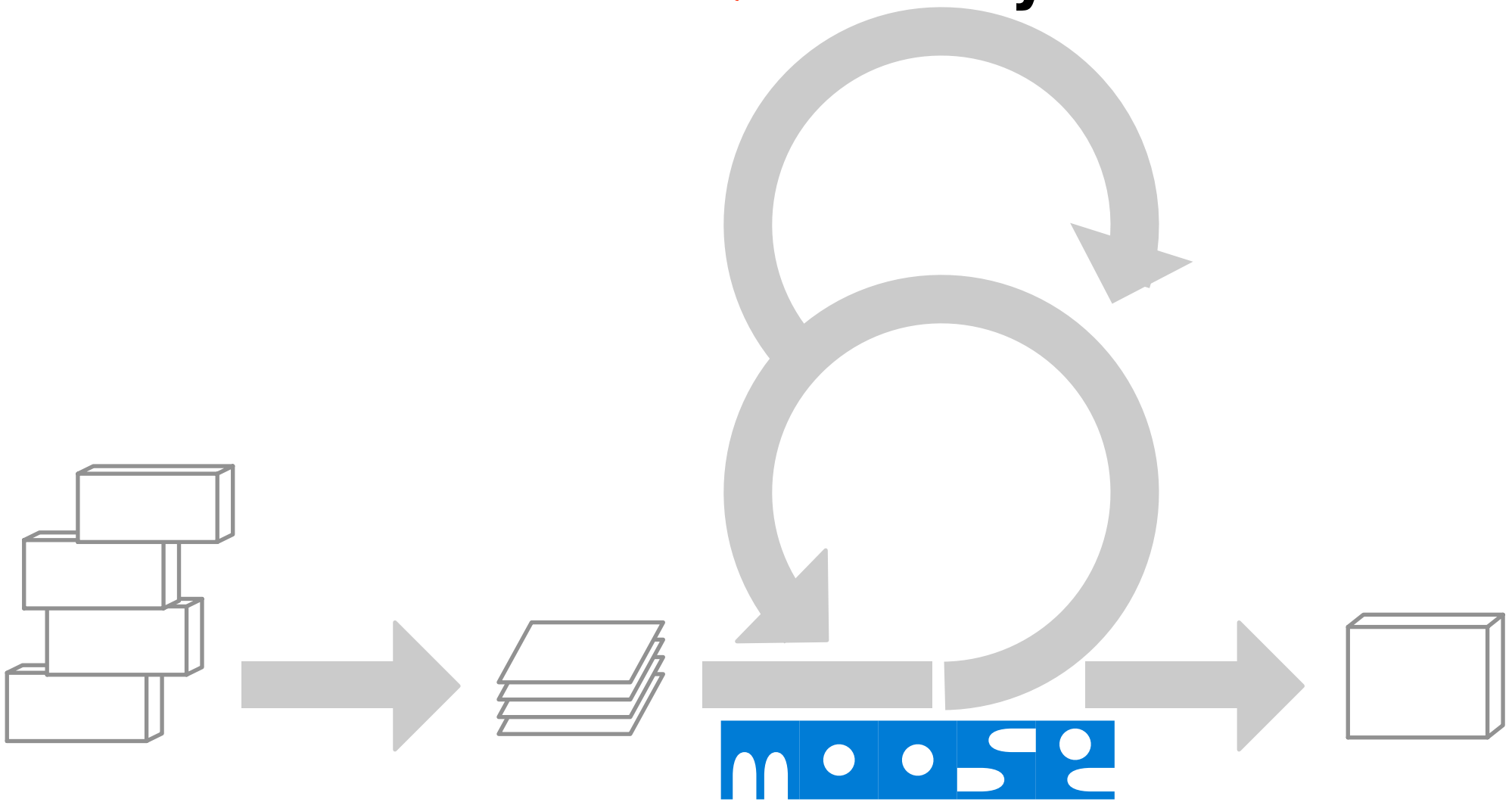



Report
Index
Remote methods called without error handling
Editor
Server
No @Interceptors in
Wrong @Interceptor
Remote methods ca
Usages of EntityMan
Direct EntityManage
Entity Beans not in p
Entities that are Ser
Embeddeable not Se
Not allowed calls to
Serializable wihtout
Serializable inner cla
Server API
TOs not serializable
TOs with non private
TOs with only static
TOs with qualified p
TOs not used
All calls to remote methods should go through an implementation of
41 remote methods
app:browser:srv:api:IFavorites
app:browser:srv:api:IFavorites
app:browser:srv:api:IHistorySe
app:browser:srv:api:IHistorySe
common:legacyentities:srv:api
common:legacyentities:srv:api
common:legacyentities:srv:api
common:legacyentities:srv:api
core:login:srv:api:ILoginServic
core:login:srv:api:ILoginServic
core:login:srv:api:ILoginServic
core:login:srv:api:ISystemMes
core:programmenu:srv:api:IPro
srv:common:service:api:IPrope
srv:common:service:api:IServe
srv:masterdata:api:IMasterDat
srv:masterdata:api:IMasterDat
srv:masterdata:api:IMasterDat
srv:masterdata:api:IMasterDat
srv:masterdata:api:IMasterDat
srv:masterdata:api:IMasterDat
getHierarchy
run
getChildElements
getElements
getChildElement
getChildElements
getChildElements
getChildren
getChildElement
getVersionisedMDHNode
getLEPNode
getDuration
derivePrescriptionDTO
createLebensbereichCategory
lepToLerf
getHierarchy
getCatalogItemTO
completeSessionEntity
saveSess
getSessions
resolveCatalogItemHierarchyForEntr
validateCase
errorhandledInitialize
masterdata:api:IMasterDataService.getHierarchy()
/**
* Provides a hierarchy, mainly used for trees.
*
* @param query
* - identifies the catalog
* @return A tree of nodes matching the query.
*/
MDHNode getHierarchy(MDItemHierarchyQueryTO query);


-moose-report #48 Test Results [Hudson]
http://-moose-report/lastCompletedBuild/testReport/
Google
Hudson
search
Hudson » -moose-report » #48 » Test Results
ENABLE AUTO REFRESH
Back to Project
Status
Changes
Console Output [raw]
History
Test Result
Previous Build
Test Result
17 failures (+16)
46 tests (±0)
Took 4.5 sec.
add description
All Failed Tests
Test Name
Duration
Age
>>> Concern.(No @Interceptors in remote implementations) 0.1 1
>>> Concern.(Remote methods called without error handling) 0.1 1
>>> Concern.(Usages of EntityManager) 0.1 1
>>> Concern.(Direct EntityManager.persist() calls) 0.1 1
>>> Concern.(Entities that are Serializable) 0.1 1
>>> Concern.(Embeddeable not Serializable) 0.1 1
>>> Concern.(TOs with only static attributes) 0.1 1
>>> Concern.(TOs not used) 0.1 1
>>> Concern.(Remote interfaces incorrectly packaged) 0.1 1
>>> Concern.(Entity Beans incorrectly packaged) 0.1 1
>>> Concern.(TOs incorrectly named) 0.1 1
>>> Concern.(Jobs scheduled without progressbar) 0.1 1
>>> Concern.(Usages of StringBuffer) 0.1 1
>>> Concern.(Usages of java::util::Vector) 0.1 1
>>> Concern.(Usages of setTimeInMillis) 0.1 1
>>> Concern.(Attributes not following naming convention) 0.1 1
>>> Concern.(Serializable wihtout default constructor) 0.1 15

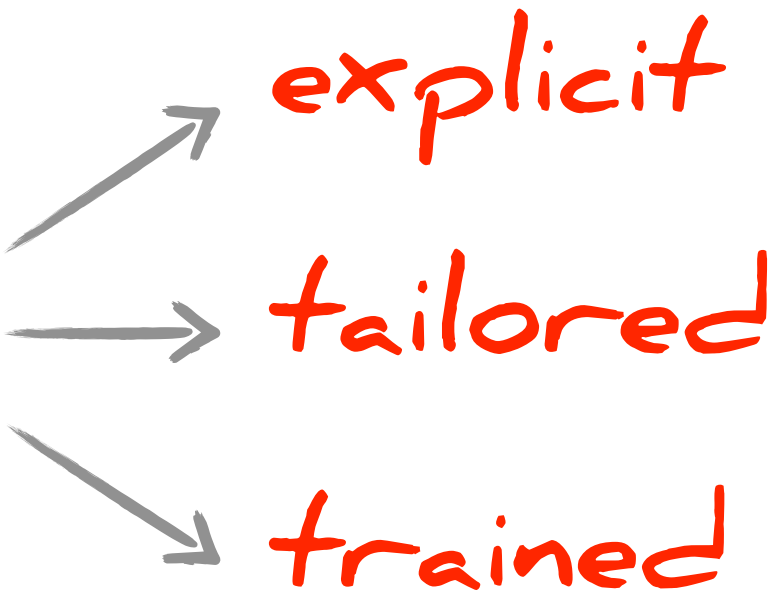

-moose-report #48 Test Results [Hudson]
http://
-moose-report/lastCompletedBuild/testReport/
Google
Hudson
search
Hudson » -moose-report » #48 » Test Results
ENABLE AUTO REFRESH
Back to Project
Status
Changes
Console Output [raw]
History
Test Result
Previous Build
Test Result
17 failures (+16)
46 tests (±0)
Took 4.5 sec.
add description
All Failed Tests
Test Name
Duration
Age
>>> Concern.(No @Interceptors in remote implementations) 0.1 1
>>> Concern.(Remote methods called without error handling) 0.1 1
>>> Concern.(Usages of EntityManager) 0.1 1
>>> Concern.(Direct EntityManager.persist() calls) 0.1 1
>>> Concern.(Entities that are Serializable) 0.1 1
>>> Concern.(Embeddeable not Serializable) 0.1 1
>>> Concern.(TOs with only static attributes) 0.1 1
>>> Concern.(TOs not used) 0.1 1
>>> Concern.(Remote interfaces incorrectly packaged) 0.1 1
>>> Concern.(Entity Beans incorrectly packaged) 0.1 1
>>> Concern.(TOs incorrectly named) 0.1 1
>>> Concern.(Jobs scheduled without progressbar) 0.1 1
>>> Concern.(Usages of StringBuffer)
>>> Concern.(Usages of java::util::Vector)
>>> Concern.(Usages of setTimeInMillis)
>>> Concern.(Attributes not following naming
>>> Concern.(Serializable
15
continuous assessment

feedback is key



specific feedback is better

Humane
assessment must be



Tudor Gîrba
www.tudorgirba.com

Stéphane Ducasse
stephane.ducasse.free.fr