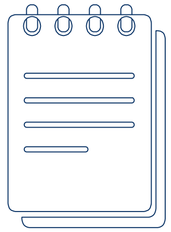
Software Architecture Quality Case Studies on Open Source Projects

Ural Erdemir & Sinan Eski
@ **CodeMR**



CONTENT

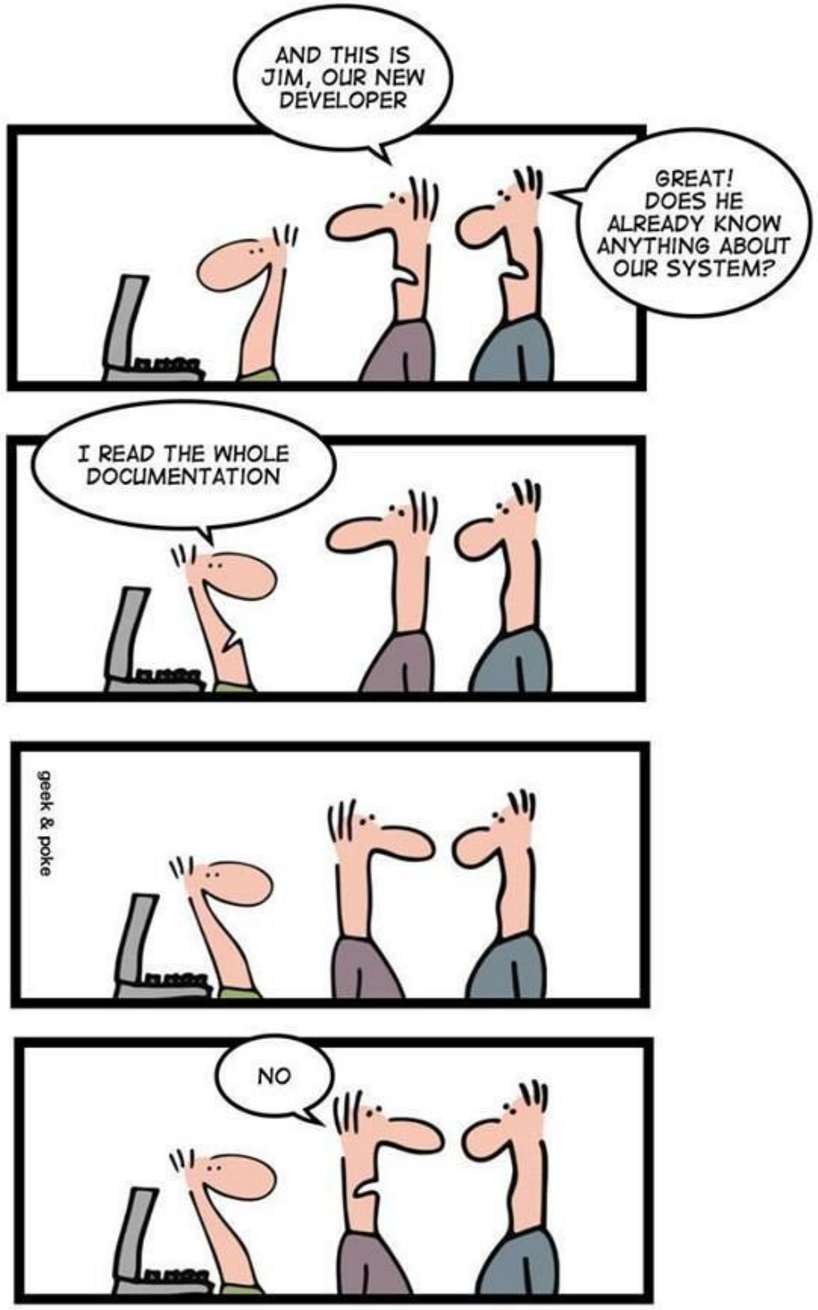
- Software Design & Principles
- What is software quality?
- Why software quality?
- Technical Debt
- How to measure software quality?
- Case Studies on Open Source Projects

SOFTWARE DESIGN & PRINCIPLES

Readable
Extendable
Maintainable

“**Maintenance** typically claims **40-80%** of all projects cost”

Barry Boehm



- Design Documents
  - Outdated
  - Incomplete
  - Absent
- High rate of turnover among developers

Developers spend 80% of development time reading code!

Funny comments by previous developers

```
//When I wrote this, only God and I understood what I was doing
//Now, God only knows
```

```
//
// Dear maintainer:
//
// Once you are done trying to 'optimize' this routine,
// and have realized what a terrible mistake that was,
// please increment the following counter as a warning
// to the next guy:
//
// total_hours_wasted_here = 42
//
```

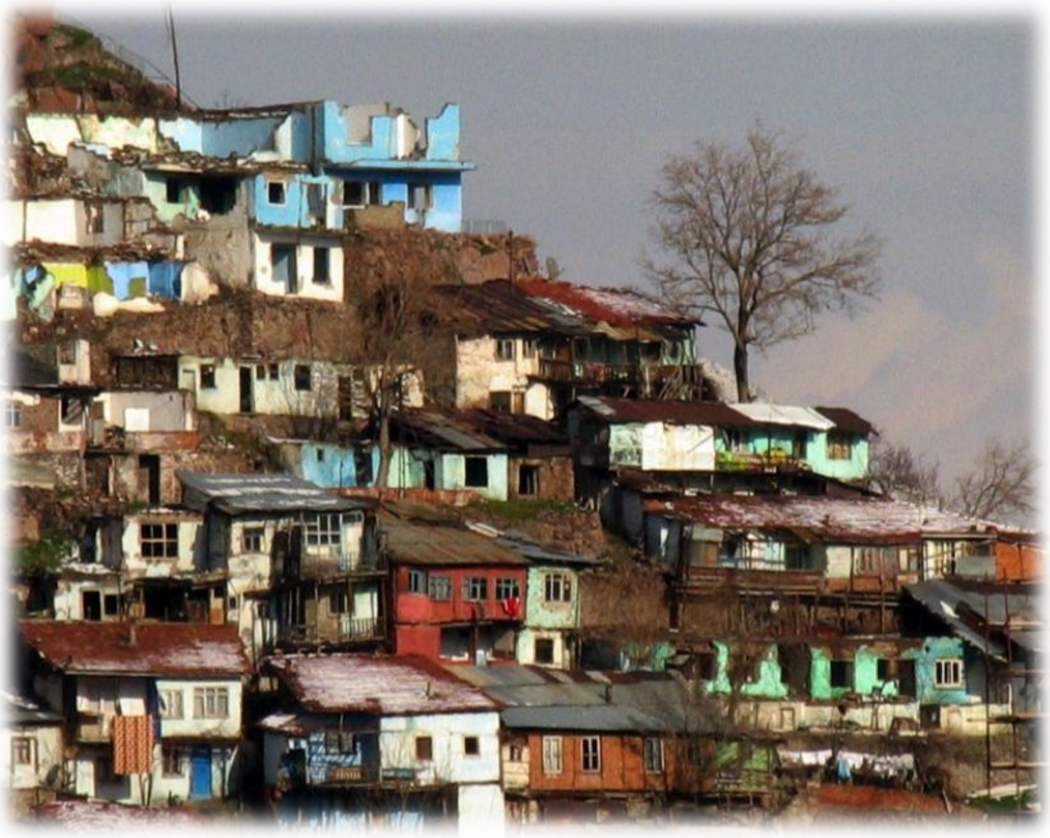
```
/*
 * You may think you know what the following code does.
 * But you dont. Trust me.
 * Fiddle with it, and youll spend many a sleepless
 * night cursing the moment you thought youd be clever
 * enough to "optimize" the code below.
 * Now close this file and go play with something else.
 */
```

the source code determines what the actual design really is

software tools would be available that post processed a source code design and generated the auxiliary documentation.

What is Software Design?
Jack W. Reeves ©1992 C++ Journal

Good vs Bad Design



Design Principles & Patterns

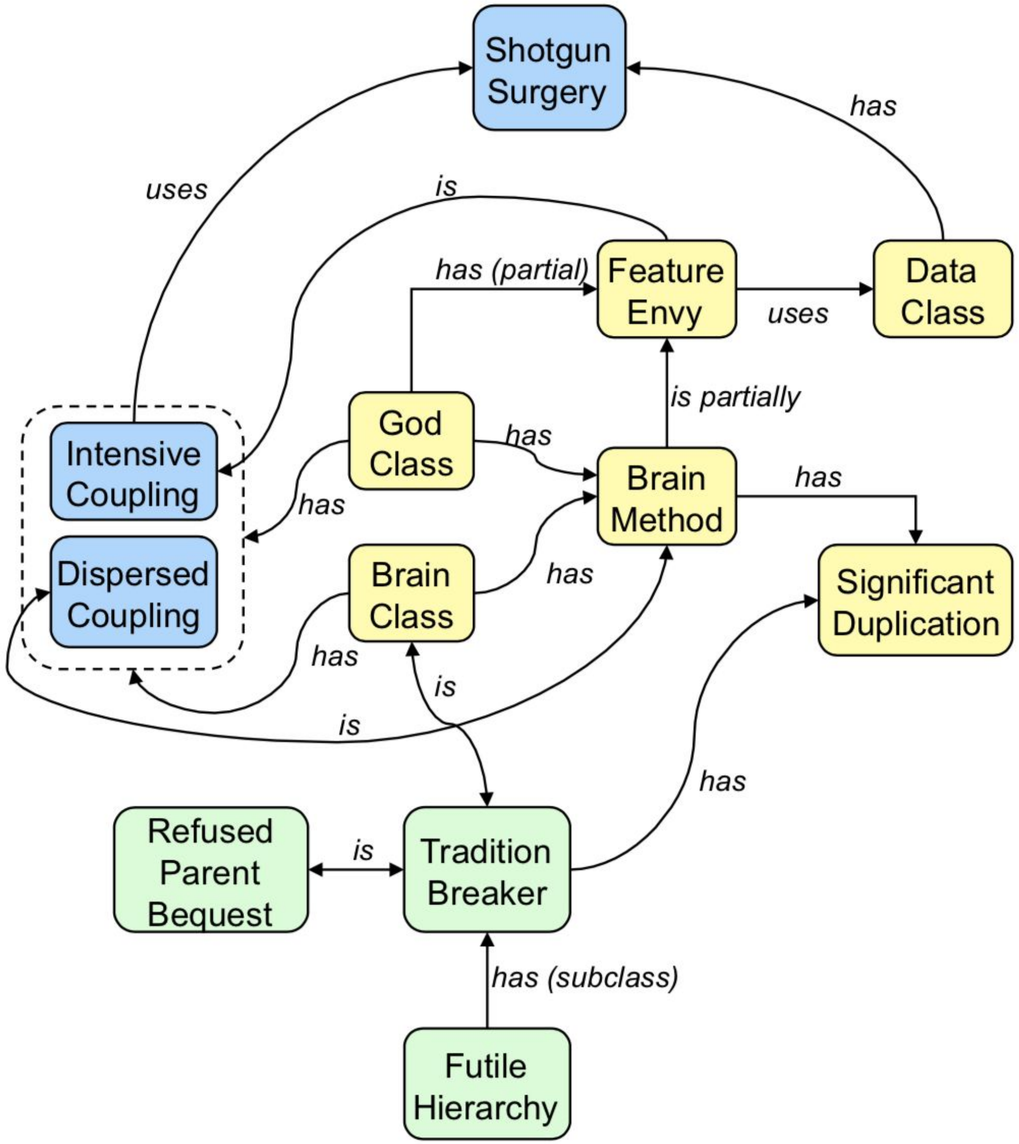
SOLID
YAGNI
DRY

GoF DESIGN PATTERNS

Code Smells

"**a code smell is a surface indication that usually corresponds to a deeper problem in the system**".

Martin Fowler



Identity Disharmonies

Collaboration Disharmonies

Classification Disharmonies

Object-Oriented Metrics in Practice

by Michele Lanza, Radu Marinescu

WHAT IS SOFTWARE QUALITY AND WHY DO WE NEED IT ?

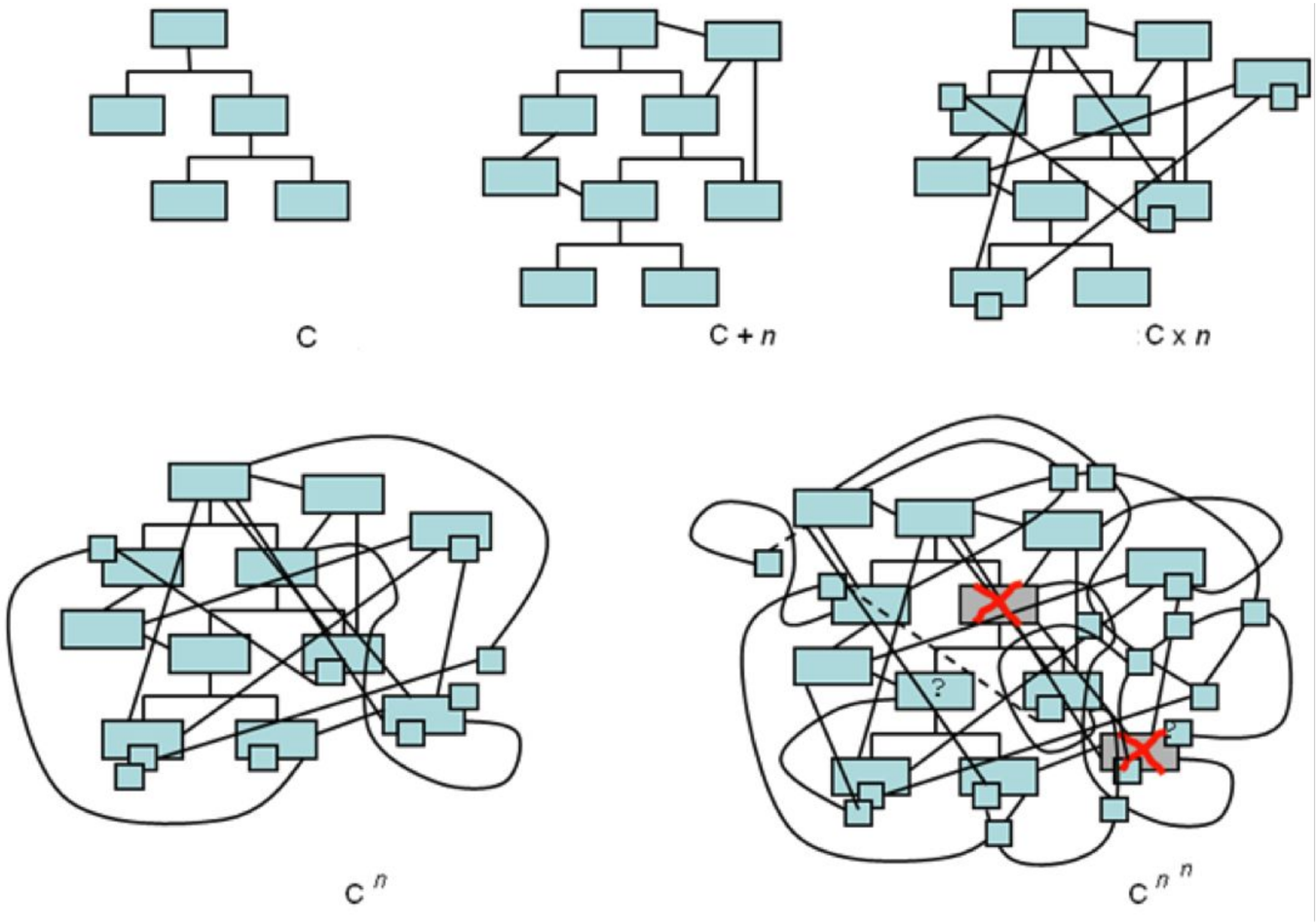
Functionality
Performance
Reliability
Usability...
Coupling
Cohesion
Complexity
Design Structure
Testability
Reusability
...

Poor Quality

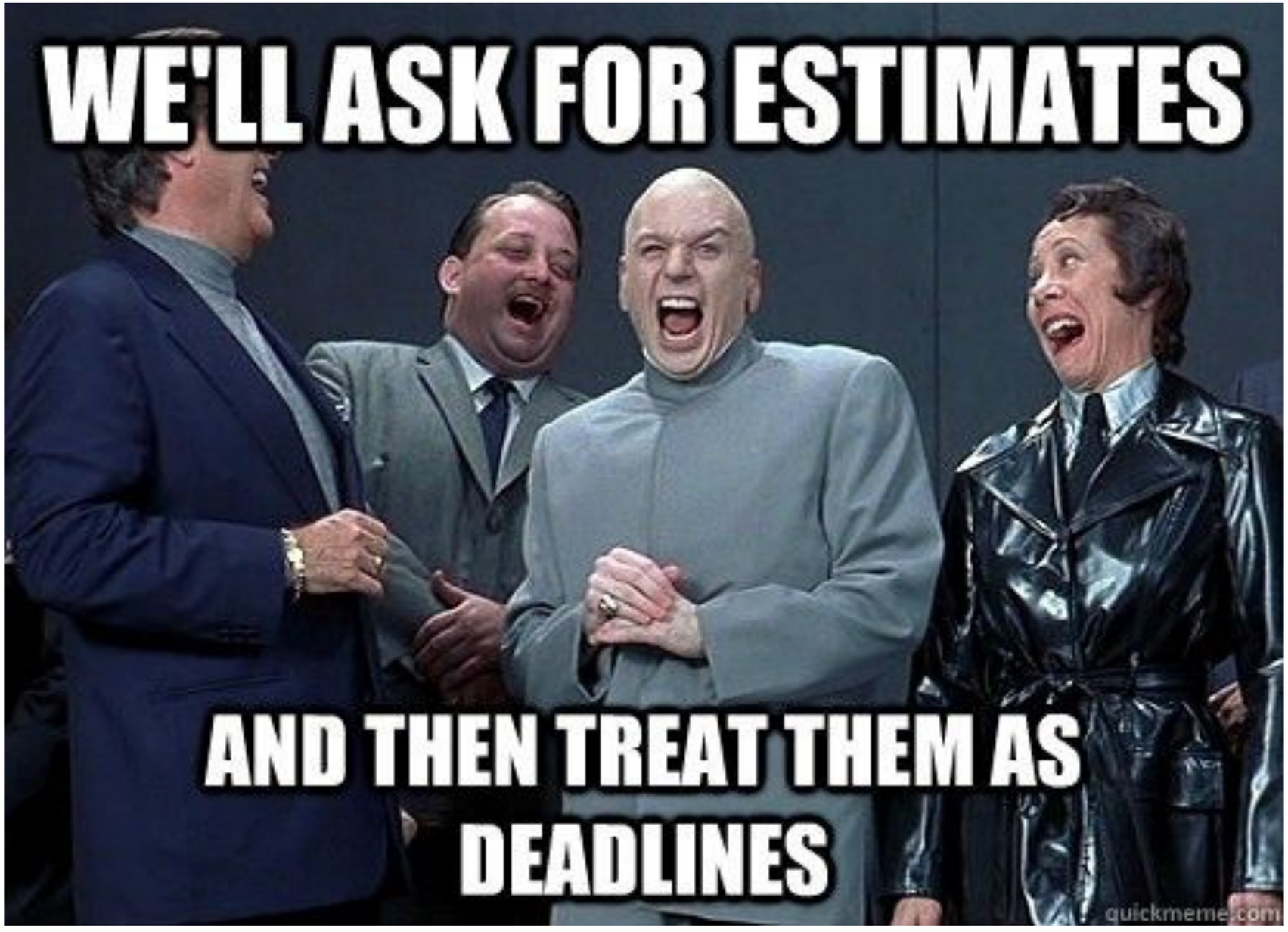
Rio de Janeiro. 3am, 2 of February, 1998 a building called Palace II starts to crumble. The building was finished less than 3 years before that. 44 apartments were completely destroyed, and 8 people died.

Many experts said that the concrete had too much water and contained beach sand. Also, they said that there wasn't an engineer overlooking the construction, and that there was basically no quality control.

Relation Between Design Quality & Change Cost



REASONS


WE'LL ASK FOR ESTIMATES
AND THEN TREAT THEM AS DEADLINES
quickmeme.com

“When, due to constraints, I design *quickly and dirty*, my project is loaded with technical debt”

Ward Cunningham, 1992

TechnicalDebt

Martin Fowler

1 October 2003

You have a piece of functionality that you need to add to your system. You see two ways to do it, one is quick to do but is messy - you are sure that it will make further changes harder in the future. The other results in a cleaner design, but will take longer to put in place.

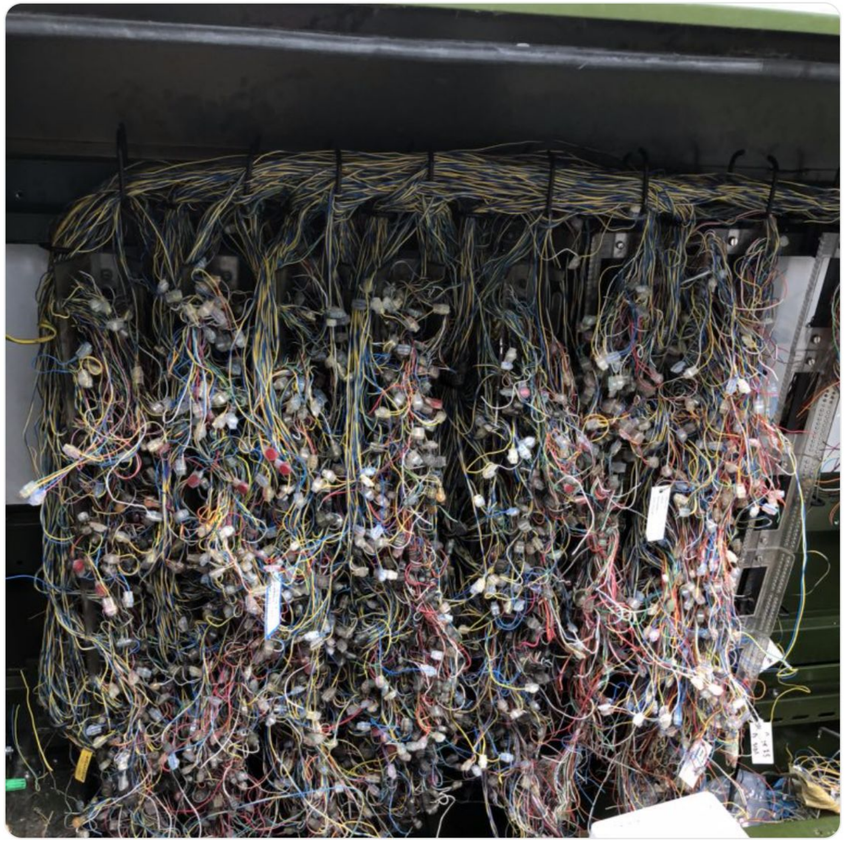
Technical Debt is a wonderful metaphor developed by Ward Cunningham to help us think about this problem. In this metaphor, doing things the quick and dirty way sets us up with a technical debt, which is similar to a financial debt. Like a financial debt, the technical debt incurs interest payments, which come in the form of the extra effort that we have to do in future development because of the quick and dirty design choice. We can choose to continue paying the interest, or we can pay down the principal by refactoring the quick and dirty design into the better design. Although it costs to pay down the principal, we gain by reduced interest payments in the future.

https://martinfowler.com/bliki/TechnicalDebt.html

Kent Beck
@KentBeck

Following

Technical debt

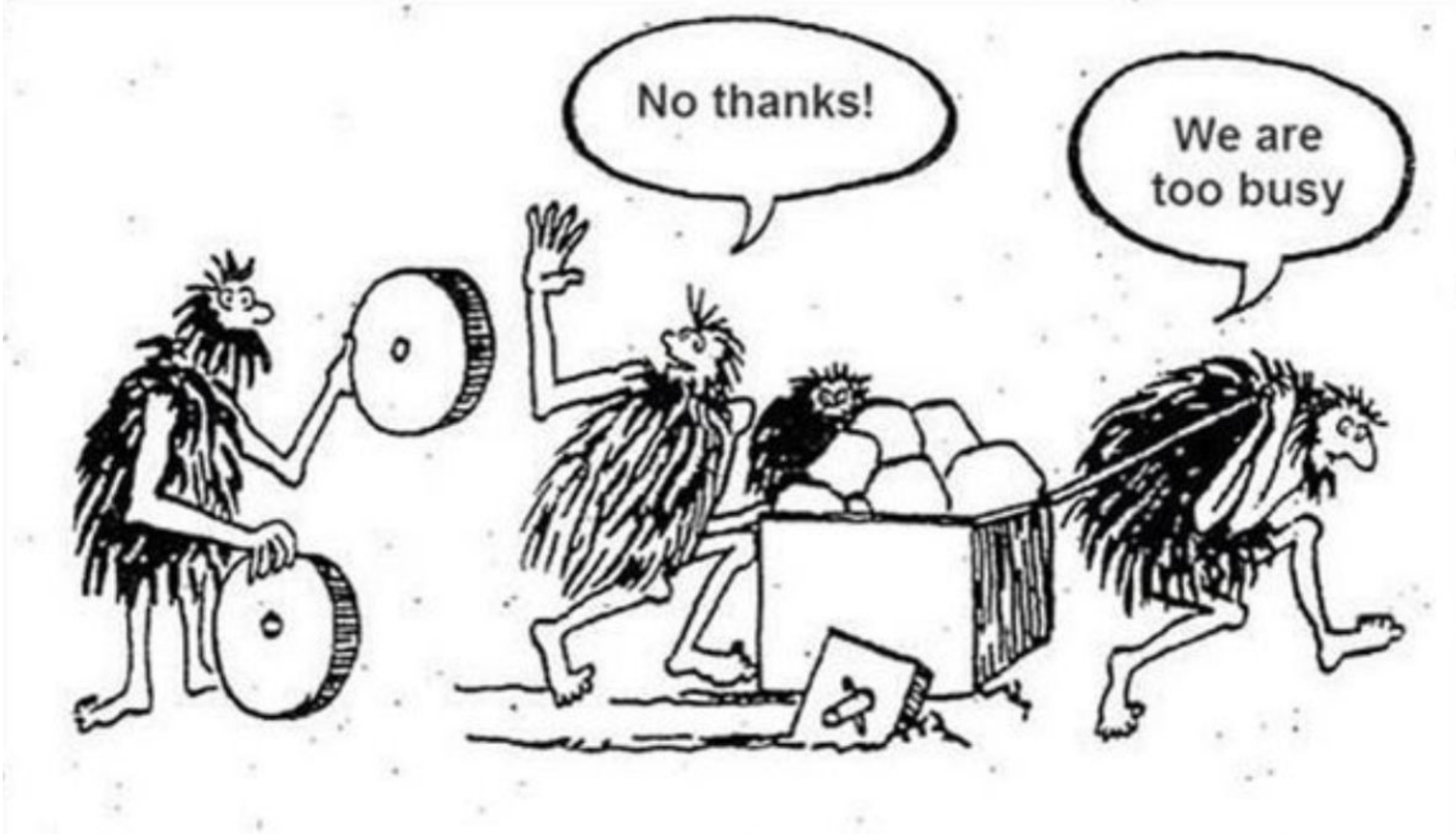


3:25 AM - 5 Oct 2018

1,762 Retweets **3,752** Likes

92 1.8K 3.8K

Technical Debt



BROKEN WINDOW THEORY

James Q. Wilson and George Kelling

“Prevention is better than cure”

Desiderius Erasmus

The cost of a defect rises significantly the later in the process it is discovered.

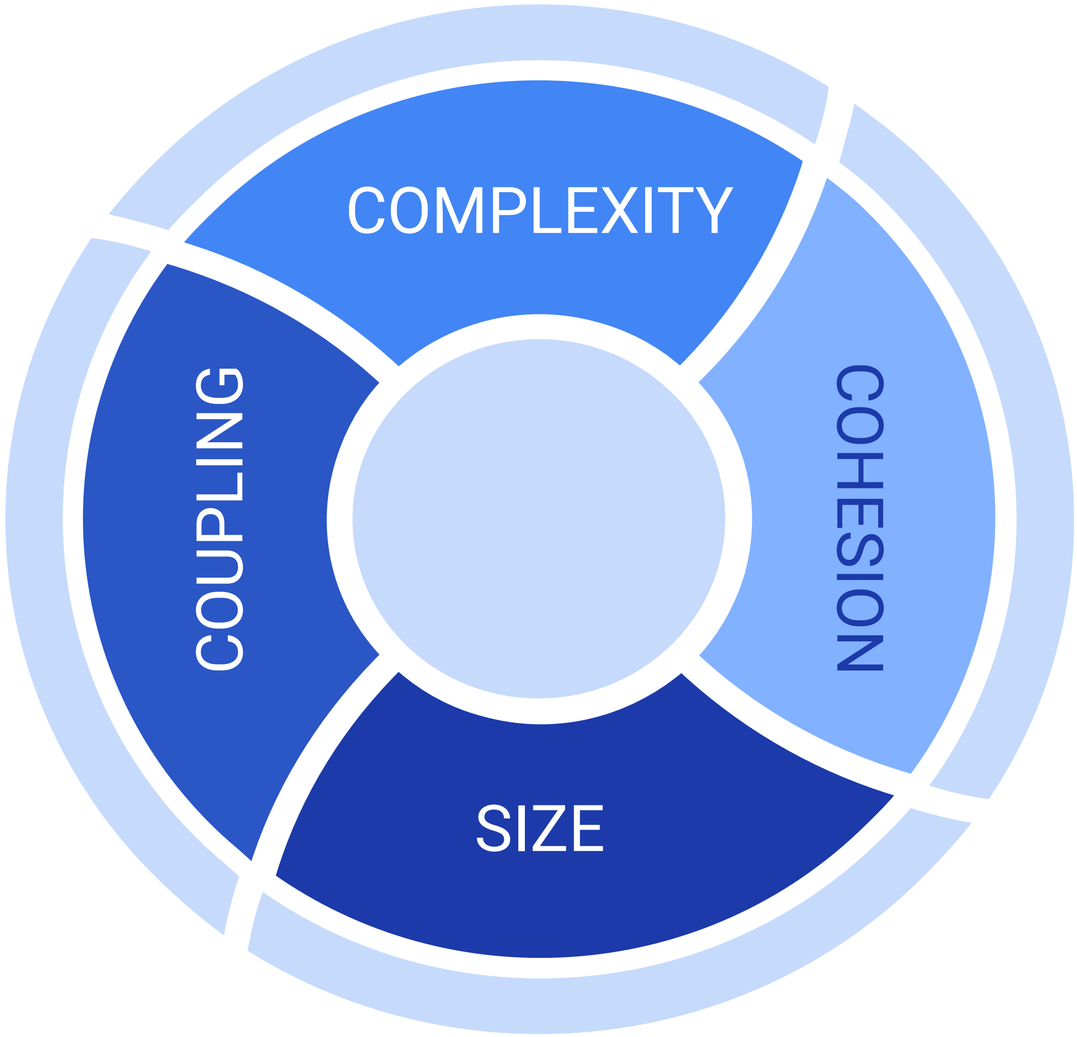
HOW TO MEASURE SOFTWARE QUALITY?

“

“You can't control what you can't measure”

Tom DeMarco

Internal Quality Attributes

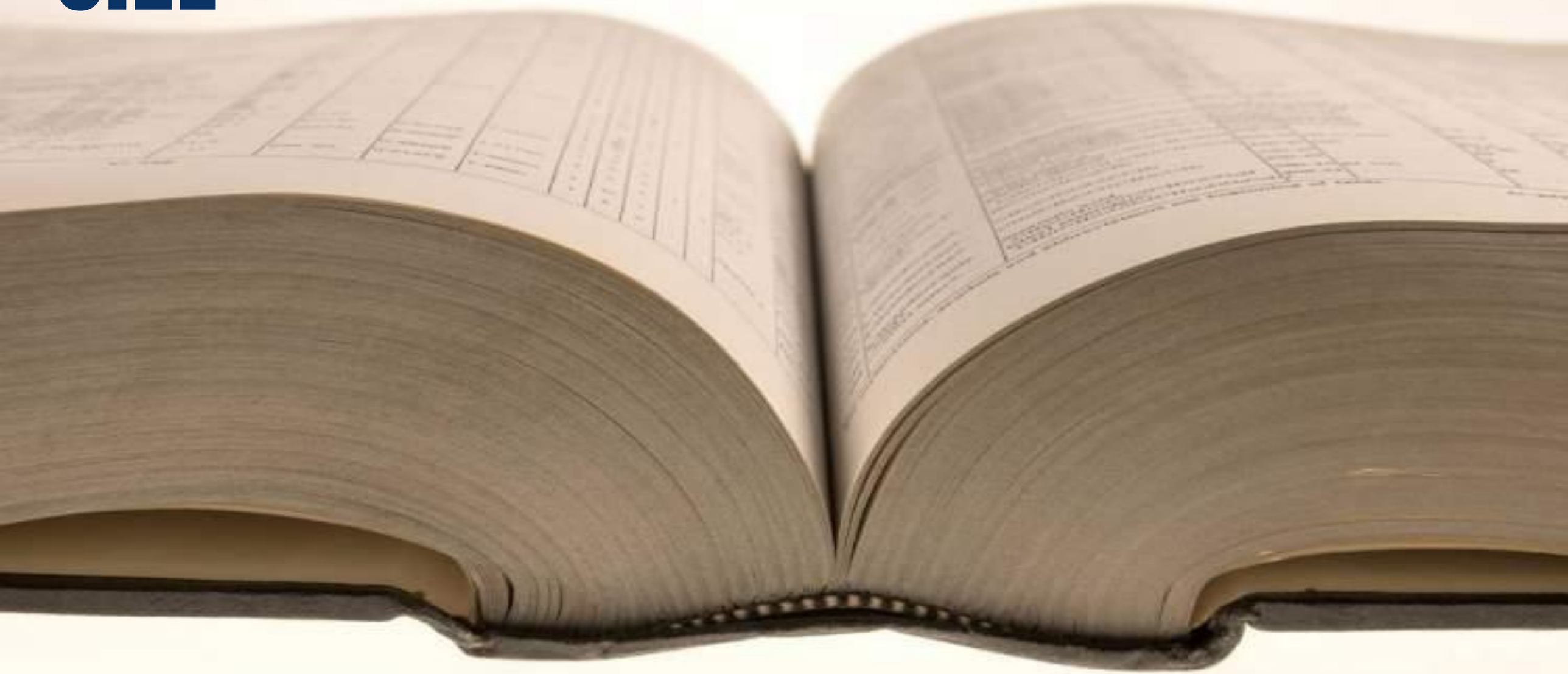


COMPLEXITY

COHESION

COUPLING

SIZE



Software metrics are quantitative measures of the degree to which a software component possesses a given attribute.

- Design Patterns have positive impact on metrics
- Creational Patterns
  - Singleton, Abstract Factory, Builder, Object Pool…
- Structural Patterns
  - Adaptor, Bridge, Facade, Composite…
- Behavioral Patterns
  - Observer, Strategy, Visitor, State…

- Code Smells/Anti-patterns have negative impact
  - Duplicate Code, Long Methods, Large Class
  - Long Parameter List, Divergent Change
  - Shotgun Surgery, Data Class, Refused Bequest
  - Dead Code…

Software Metrics

- Lines of Code (LOC),
- Number of Classes/Entities
- Number of Fields (NOF)
- Number of Methods(NOM)
- Lack of Cohesion in Methods (LCOM)
- Cohesion Amongst Methods (CAM)
- Weighted Methods per Class (WMC)
- Response For a Class (RFC)
- Depth of Inheritance Tree (DIT)
- Number of Children (NOC)
- Coupling Between Object Classes (CBO)
- Afferent/Efferent Coupling
- ...

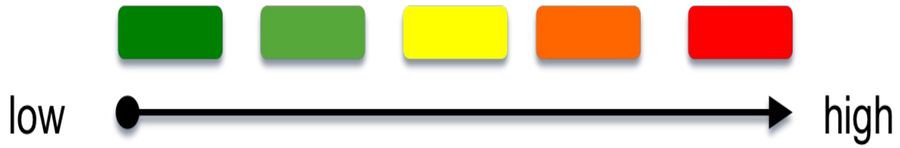
A LOT OF METRICS
AMBIGUOUS
BIG / SMALL VALUE
THRESHOLD

Why Visualization?

- Limited Time, Limited Resources
  - Which are the critical parts?
  - Where to start improvement?
- An integrated representation of complex and multi-dimensional data
  - Intuitive
  - Catchy and memorable
- Easy to perceive and comprehend
  - Lets users to see the big picture
  - Brings metrics and relationships together

Representation

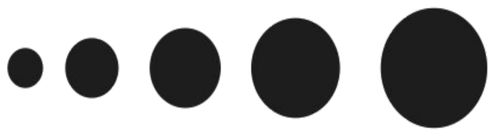
- For all metrics in charts:

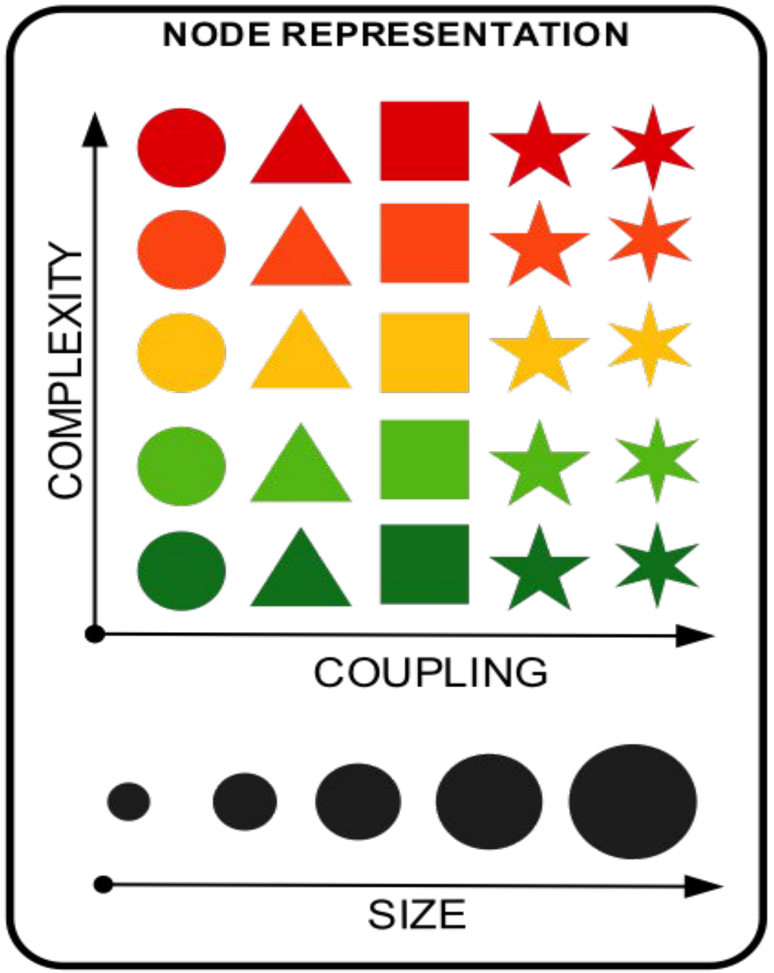


- Coupling

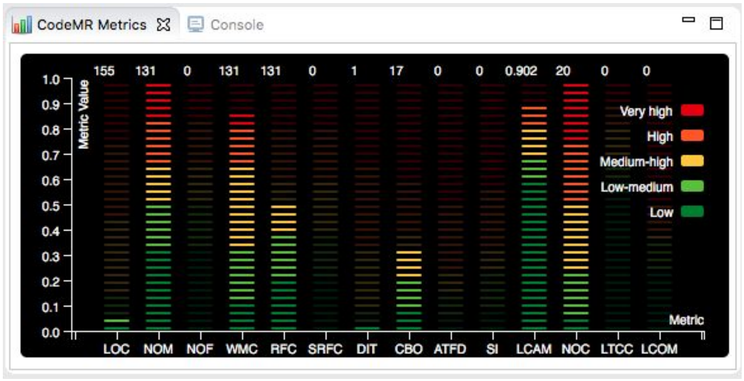
- Complexity

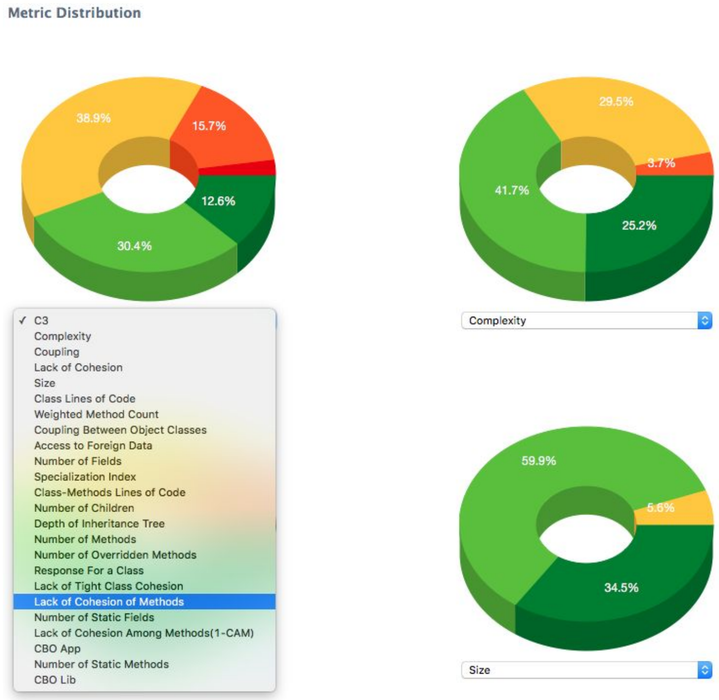
- Size





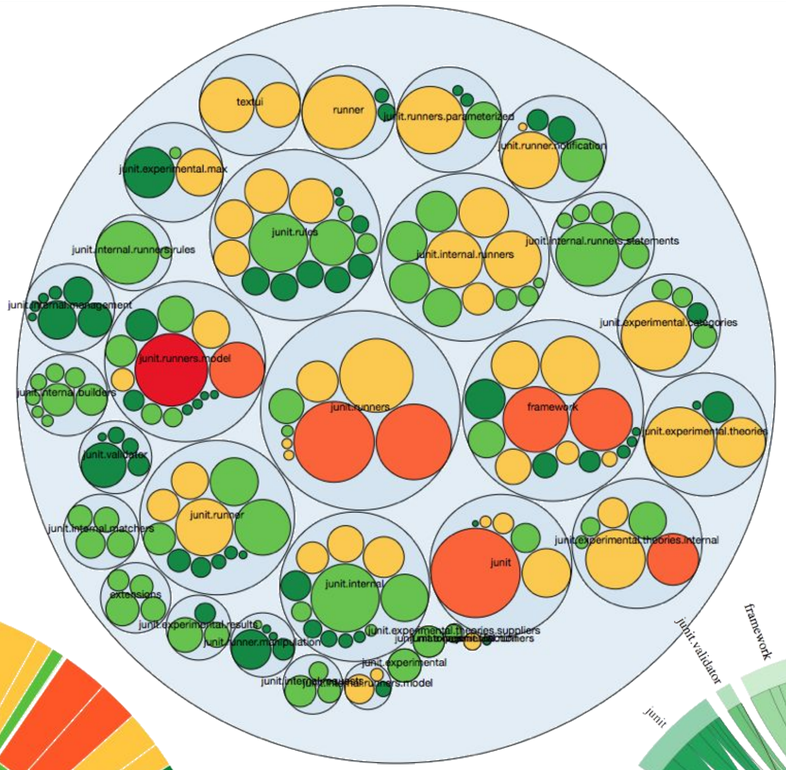
Measure Code Quality

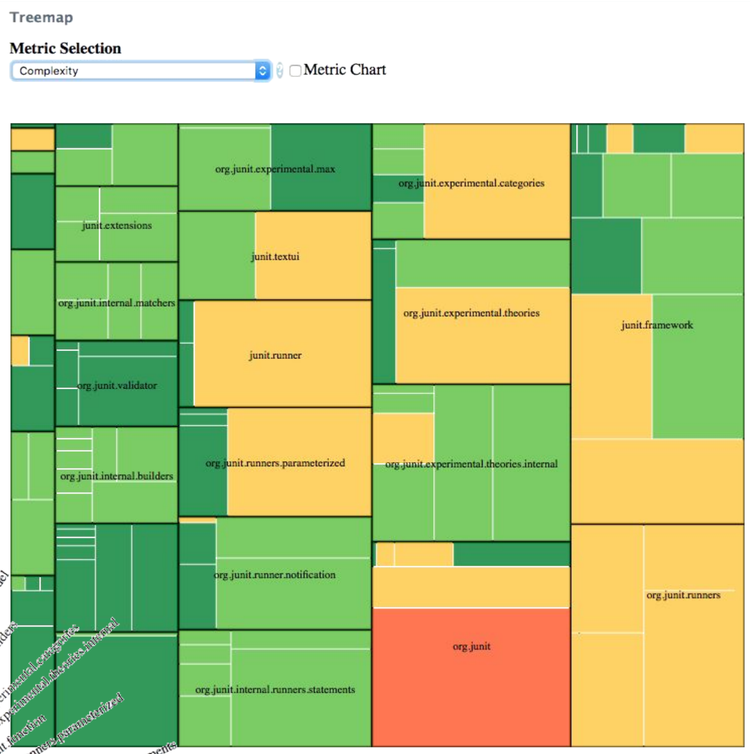


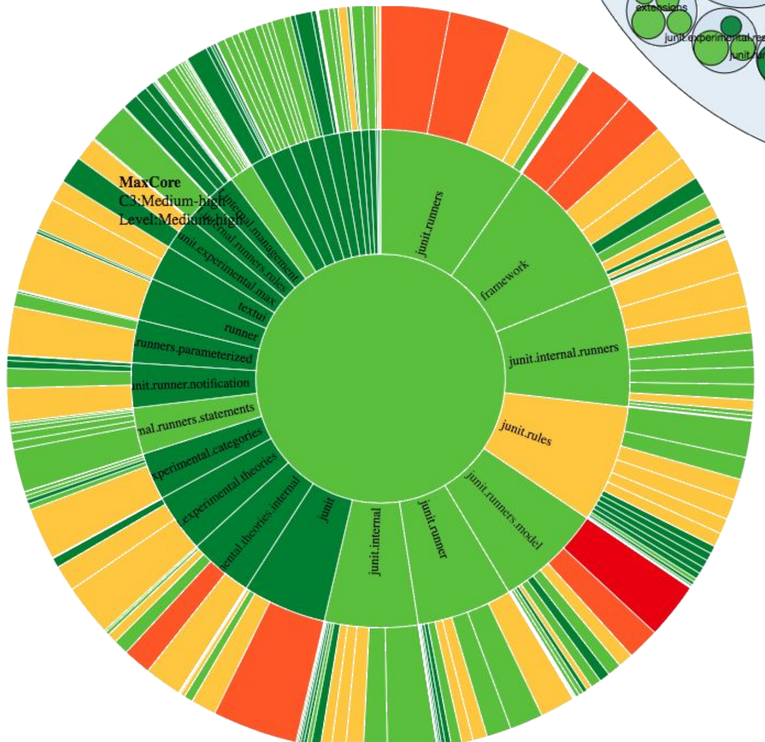


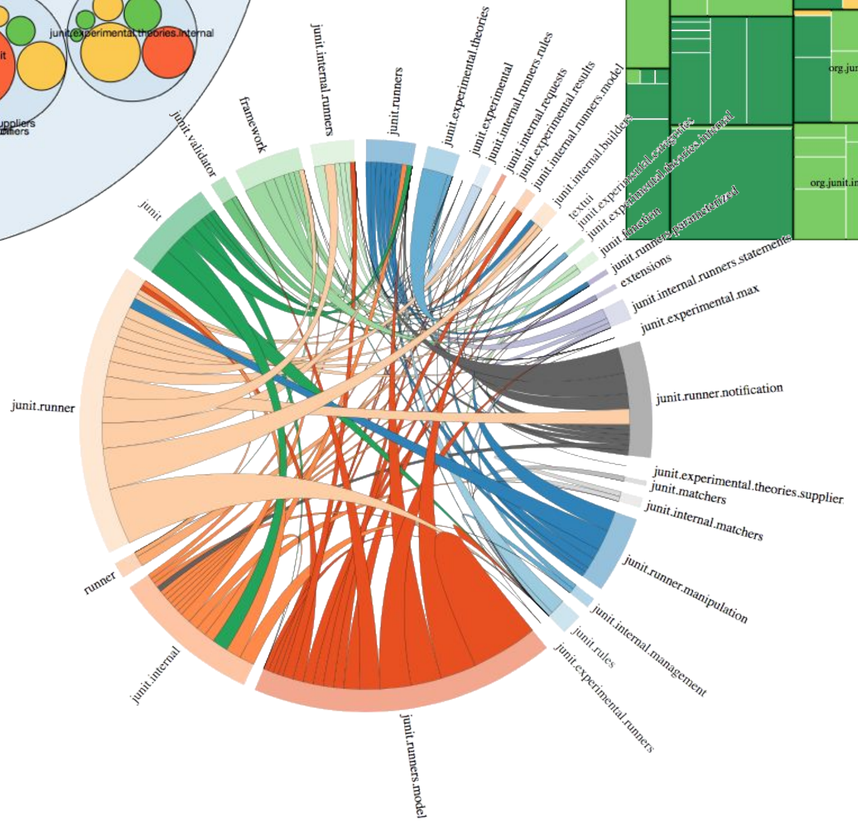
Element	Quality Attributes	LOC	Coupling	Complexity
junit4-master				
junit.framework		623	low-medium	low-medium
AssertionFailedError		7	low	medium-high
ComparisonFailure		16	low	medium-high
TestCase		132	low	medium-high
Assert		106	low	medium-high
TestResult		70	low-medium	low-medium
JUnit4TestAdapterCache		42	low-medium	low-medium
JUnit4TestAdapter		39	low-medium	low-medium
TestSuite		119	high	low-medium
TestListener		5	low	low
Protectable		2	low	low
TestFailure		19	low	low
JUnit4TestCaseFacade		14	low	low
Test		3	low	low
ComparisonCompactor		49	low	low
org.junit.runners		799	low-medium	low
org.junit.experimental.theories		348	low	low
org.junit.experimental		34	low	low
org.junit.internal.runners.rules		194	low	low
org.junit.internal.requests		67	low	low
org.junit.experimental.results		69	low	low
org.junit.internal.runners.model		38	low	low
org.junit.internal.builders		106	low-medium	low
junit.textui		153	low	low
org.junit.experimental.categories		323	low	low
org.junit.experimental.theories.interna		299	low	low
org.junit.function		2	low	low
org.junit.runners.parameterized		204	low	low
junit.extensions		83	low	low
RepeatedTest		19	low	low-medium
TestSetup		14	low	low-medium
Ljunit/extensions/TestSetup$52		0	low	low
TestSetup(Test): void		2	low	low
run(TestResult): void		9	low	low
setUp( ): void		1	low	low
tearDown( ): void		1	low	low
TestDecorator		16	low	low-medium
ActiveTestSuite		34	low	low-medium
org.junit.internal.runners.statements		235	low-medium	low
org.junit.experimental.max		182	low	low
org.junit.runner.notification		212	low	low
org.junit.experimental.theories.supplie		9	low	low

See the Big Picture

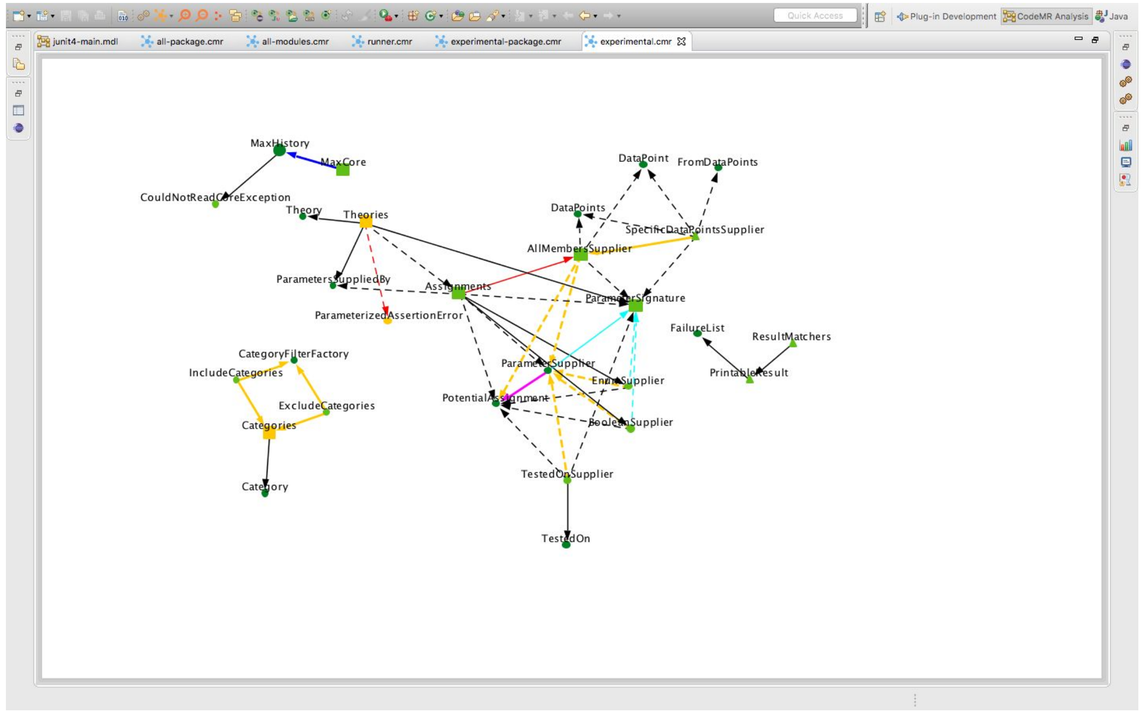


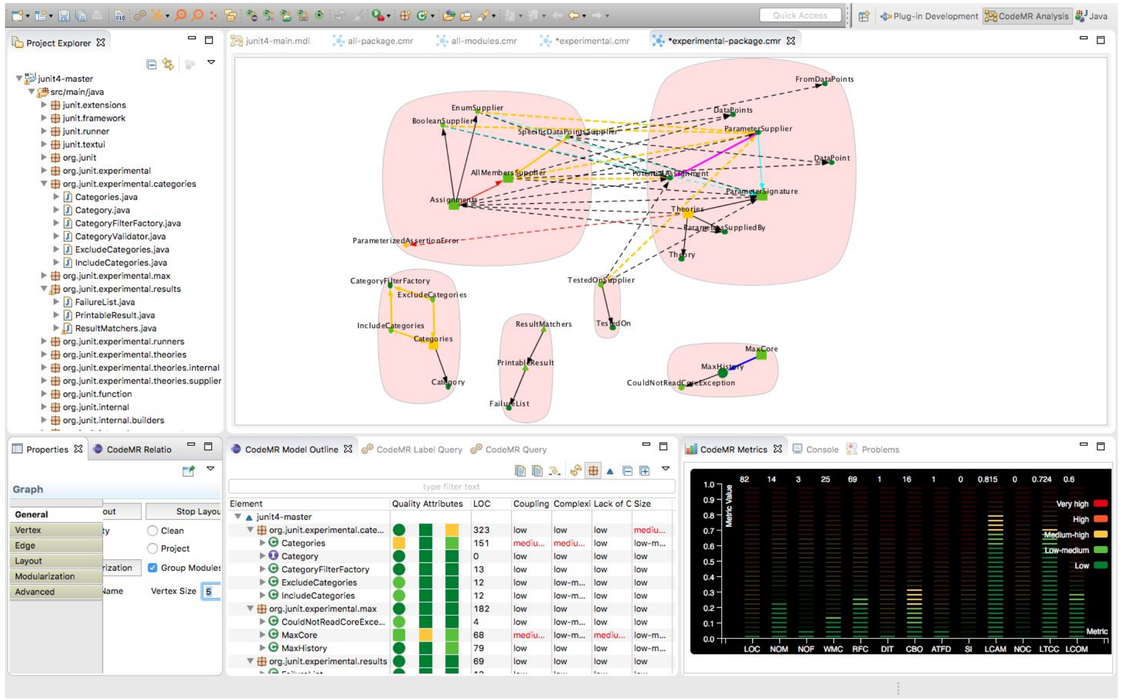






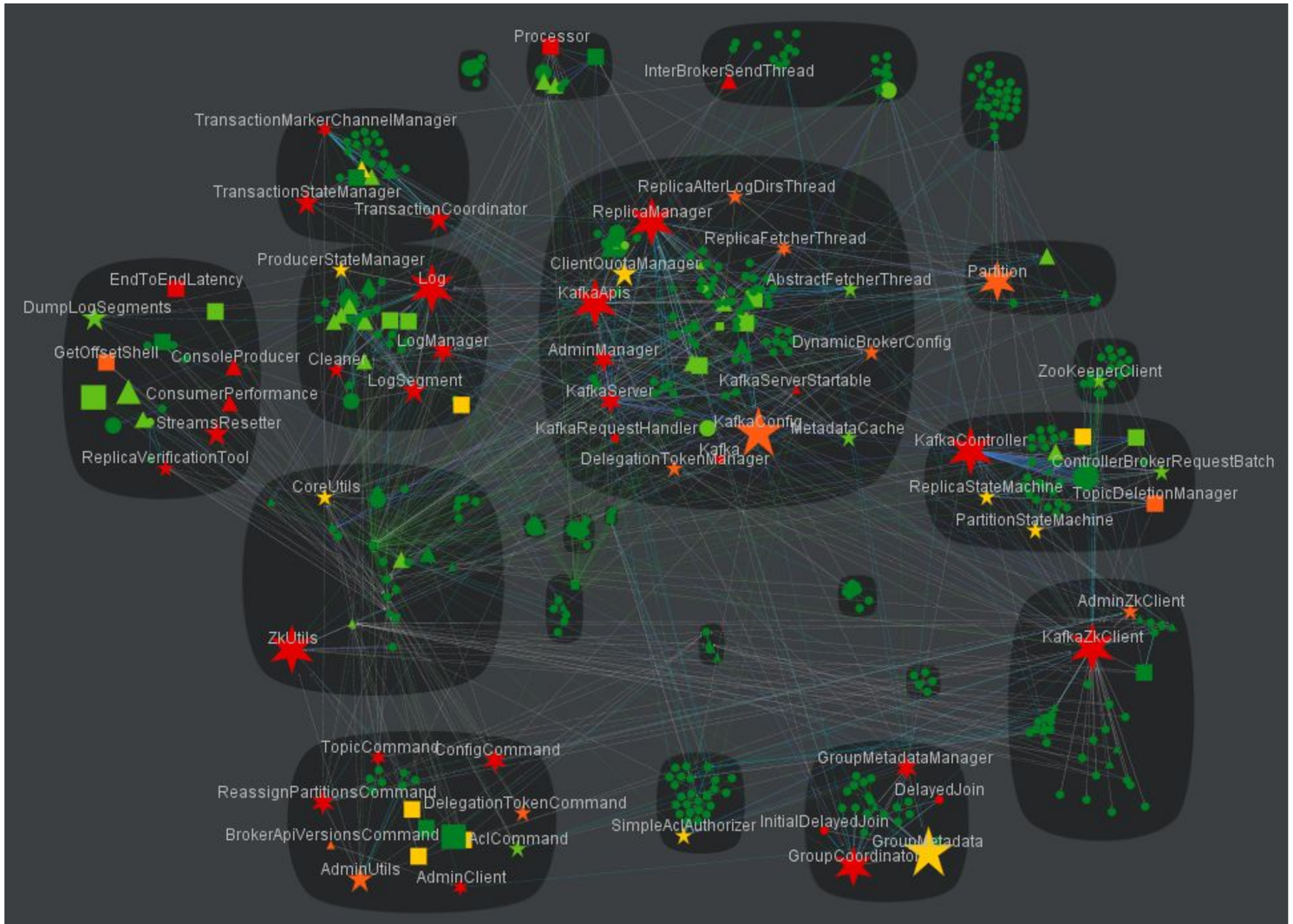
Visualization of the Relations



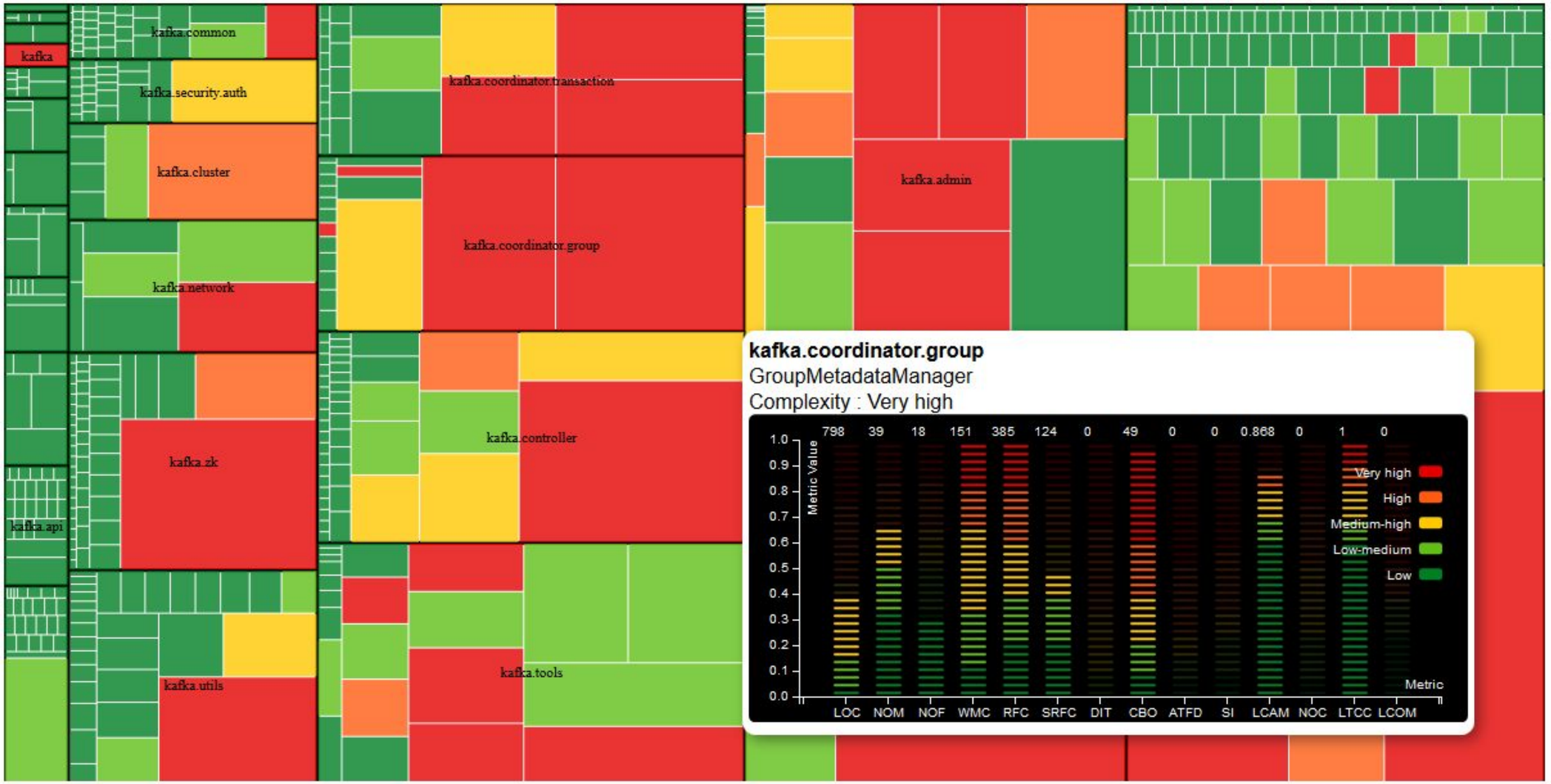


CASE STUDIES

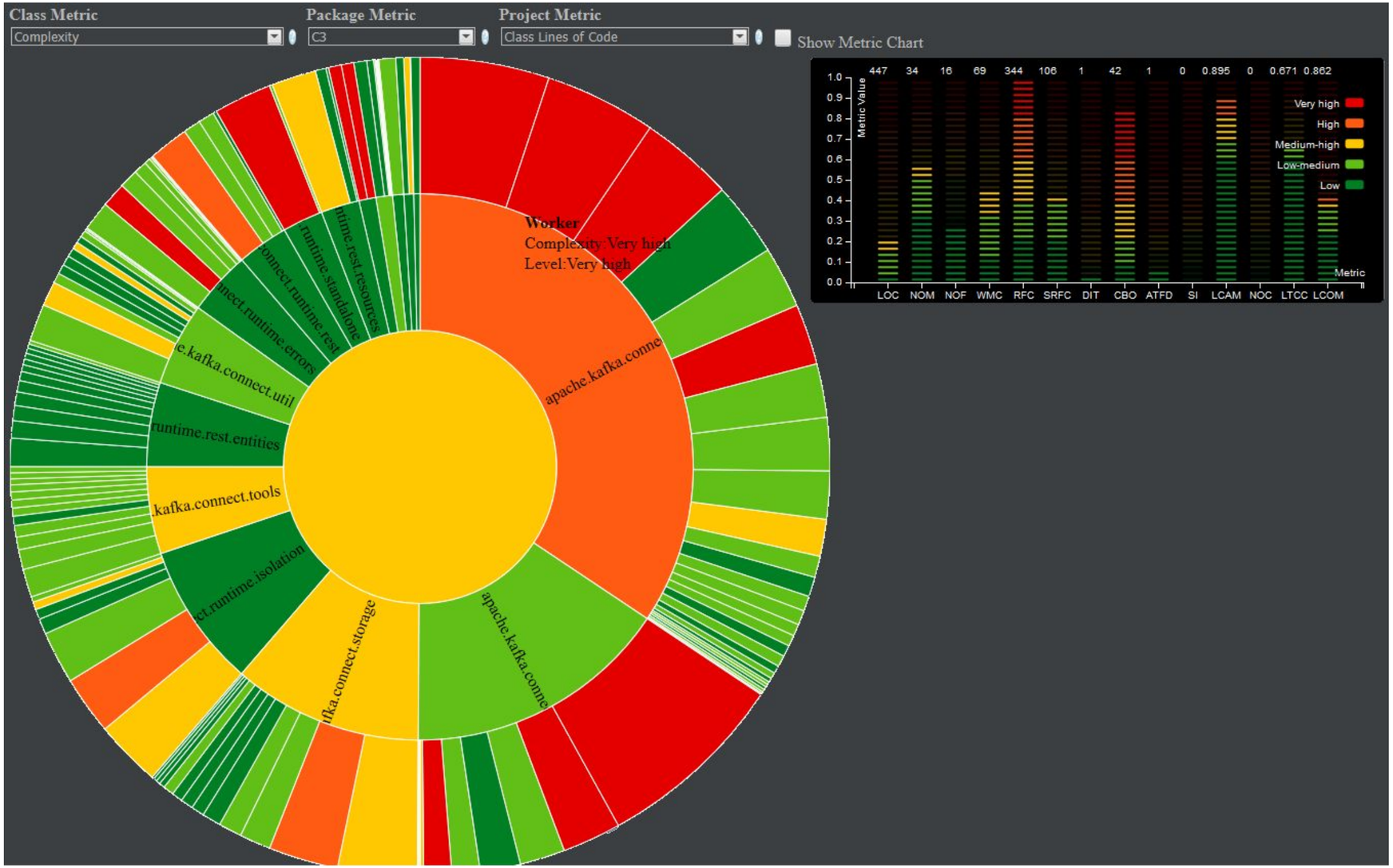
Apache Kafka



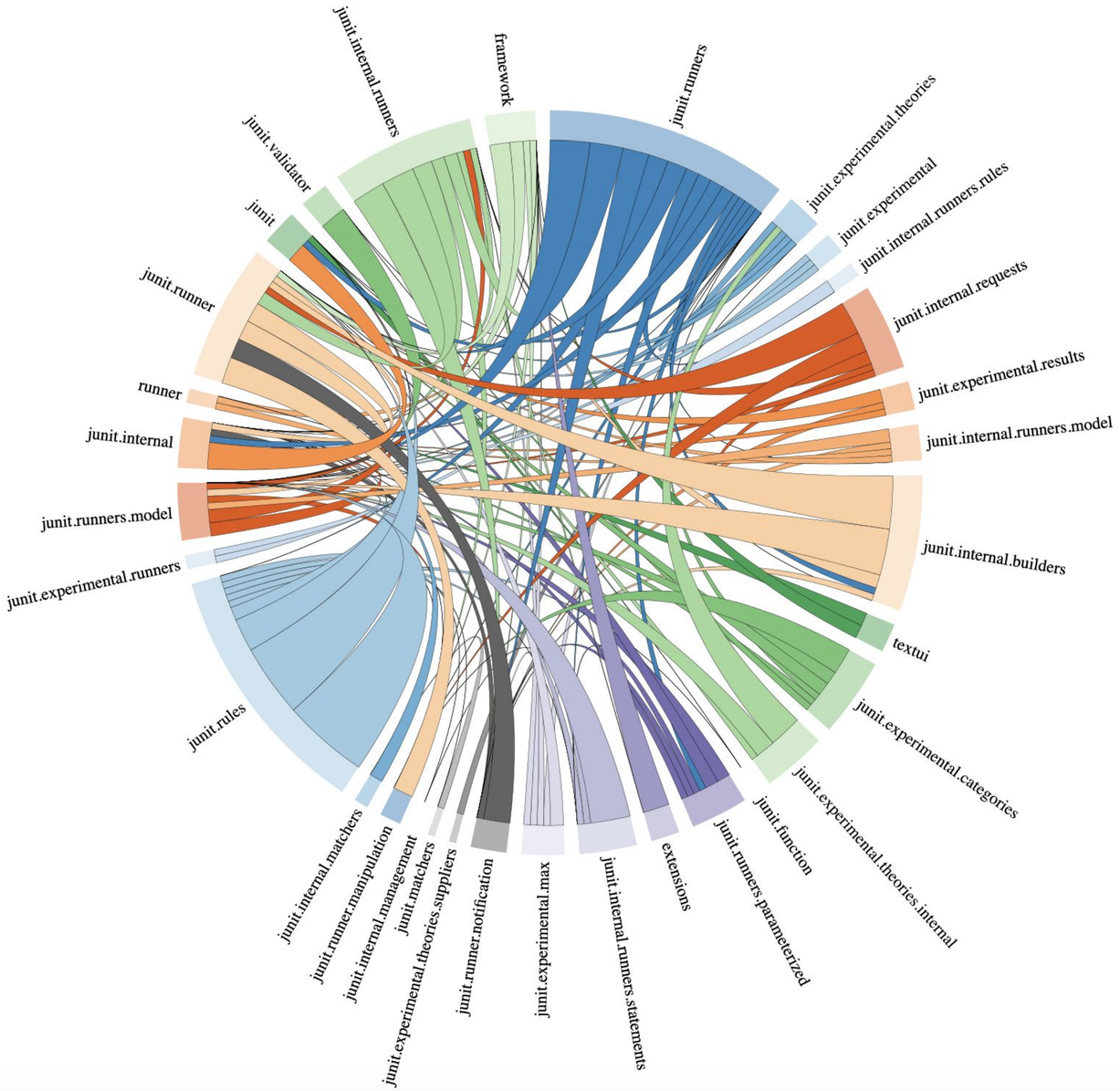
Apache Kafka



Apache Kafka



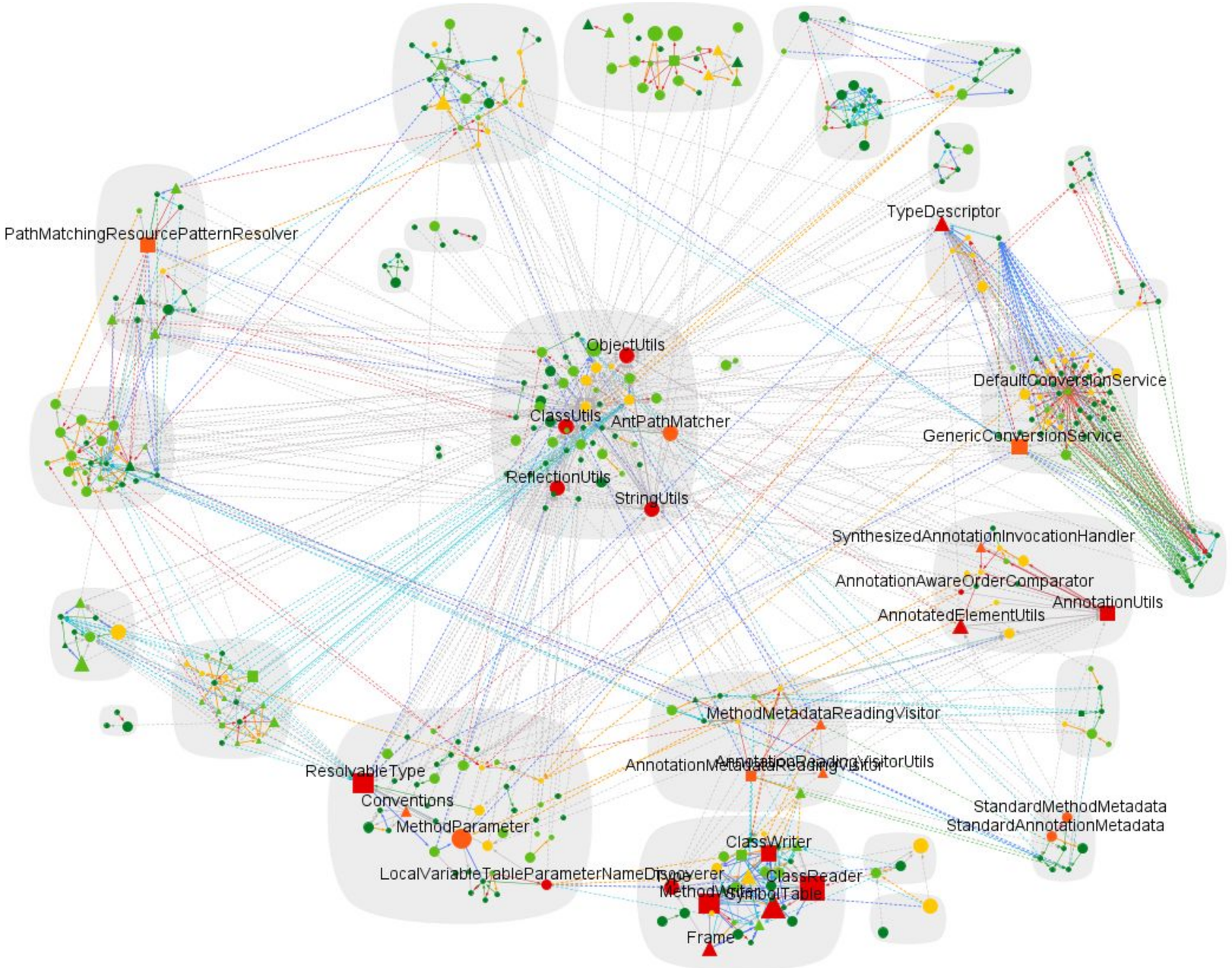
JUnit


junit.internal.runners
framework
junit.runners
junit.experimental.theories
junit.validator
junit
junit.experimental
junit.internal.runners.rules
junit.runner
junit.internal.requests
runner
junit.experimental.results
junit.internal
junit.internal.runners.model
junit.runners.model
junit.internal.builders
junit.experimental.runners
textui
junit.rules
junit.experimental.categories
junit.internal.matchers
junit.runner.manipulation
junit.internal.management
junit.matchers
junit.experimental.theories.suppliers
junit.runner.notification
junit.experimental.max
junit.internal.runners.statements
extensions
junit.runners.parameterized
junit.function
junit.experimental.theories.internal

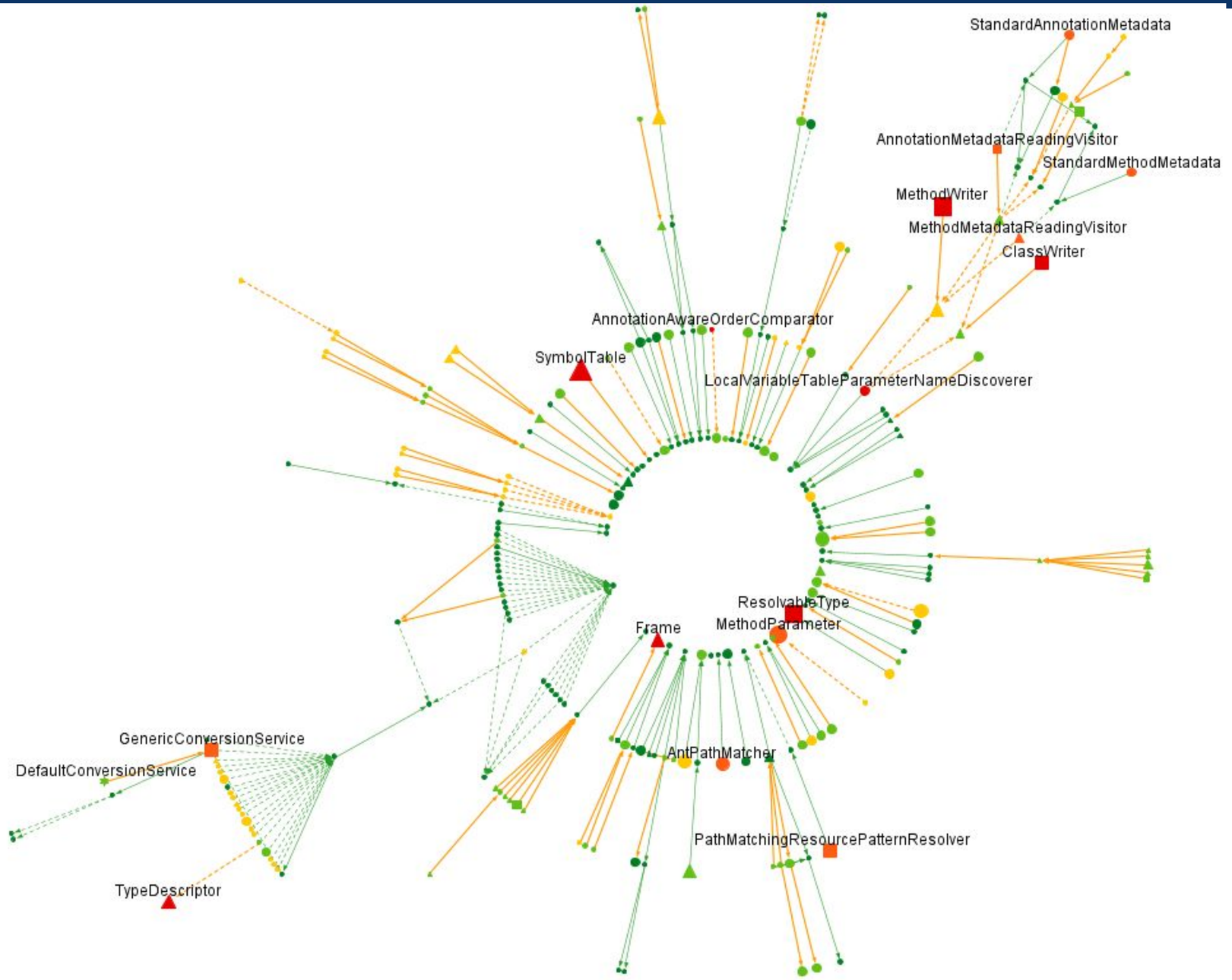
JUnit

Name	Complexity	Coupling	Size	Lack of Cohesion	CBO	RFC	SRFC	DIT	NOC	WMC	LOC	CMLOC	NOF	NOSF	NOM	N
junit4																
junit.extensions	low	low	low	low						27	83					
ActiveTestSuite	low-medium	low	low	low-medium	2	17	6	2	0	10	34	32	1	0	8	0
RepeatedTest	low-medium	low	low	low	2	11	4	3	0	7	19	17	1	0	4	0
TestDecorator	low-medium	low	low	low	1	8	3	2	2	6	16	13	1	0	6	0
TestSetup	low-medium	low	low	low	3	13	1	3	0	4	14	13	0	0	4	0
junit.framework	low-medium	low	medium-high	low						265	625					
Assert	medium-high	low	low-medium	medium-high	2	53	35	1	2	61	106	104	0	0	1	38
AssertionFailedError	medium-high	low	low	low	0	3	1	4	1	4	7	5	0	1	2	1
ComparisonCompactor	low	low	low	low	1	15	13	1	0	20	49	40	5	3	8	0
ComparisonFailure	medium-high	low	low	low	1	19	2	5	0	4	16	11	2	2	4	0
JUnit4TestAdapter	low-medium	low-medium	low	medium-high	8	56	20	1	0	16	41	37	3	0	13	0
JUnit4TestAdapterCache	low-medium	low-medium	low	low	7	21	13	3	0	11	42	39	0	2	4	1
JUnit4TestCaseFacade	low	low	low	low	2	7	2	1	0	5	14	12	1	0	5	0
Protectable	low	low	low	low	0	1	0	1	2	1	2	1	0	0	1	0
Test	low	low	low	low	1	2	0	1	5	2	3	2	0	0	2	0
countTestCases( ): int	low	low	low	low	0											
run(TestResult): void	low	low	low	low	1											
TestCase	medium-high	low	low-medium	high	1	126	55	2	1	59	132	129	1	0	13	38
TestFailure	low	low	low	low	2	11	3	1	0	7	19	16	2	0	7	0
TestListener	low	low	low	low	2	4	0	1	3	4	5	4	0	0	4	0
TestResult	low-medium	low-medium	low-medium	medium-high	6	29	19	1	0	25	70	64	5	0	18	0
TestSuite	low-medium	low	low-medium	medium-high	4	63	42	1	1	46	119	116	2	0	22	3
junit.runner	low	low	low	low						68	184					
Bas	low	low	low-medium	medium-high	5	60	46	1	1	59	169	161	1	4	16	13
Tes	low	low	low	low	0	6	0	1	0	6	9	6	0	2	6	0
Ver	low	low	low	low	0	4	2	1	0	3	6	5	0	0	1	2
junit.t	low	low	low	low						53	153					
Res	low	low	low-medium	medium-high	5	55	26	1	0	23	61	58	2	0	17	0
Tes	low	low	low-medium	medium-high	5	103	27	2	0	30	92	87	1	3	14	4
org.ju	low	low	medium-high	low						152	405					
Aft	low	low	low	low	0	0	0	1	0	0	3	0	0	0	0	0
Aft	low	low	low	low	0	0	0	1	0	0	3	0	0	0	0	0
Ass	low	low-medium	low-medium	high	6	107	64	1	0	106	252	251	0	0	1	70
Ass	low	low	low	low-medium	3	16	10	1	0	13	26	25	0	0	1	9
Ass	medium-high	low	low	low	1	4	0	5	0	4	13	10	0	1	4	0
Bef	low	low	low	low	0	0	0	1	0	0	3	0	0	0	0	0

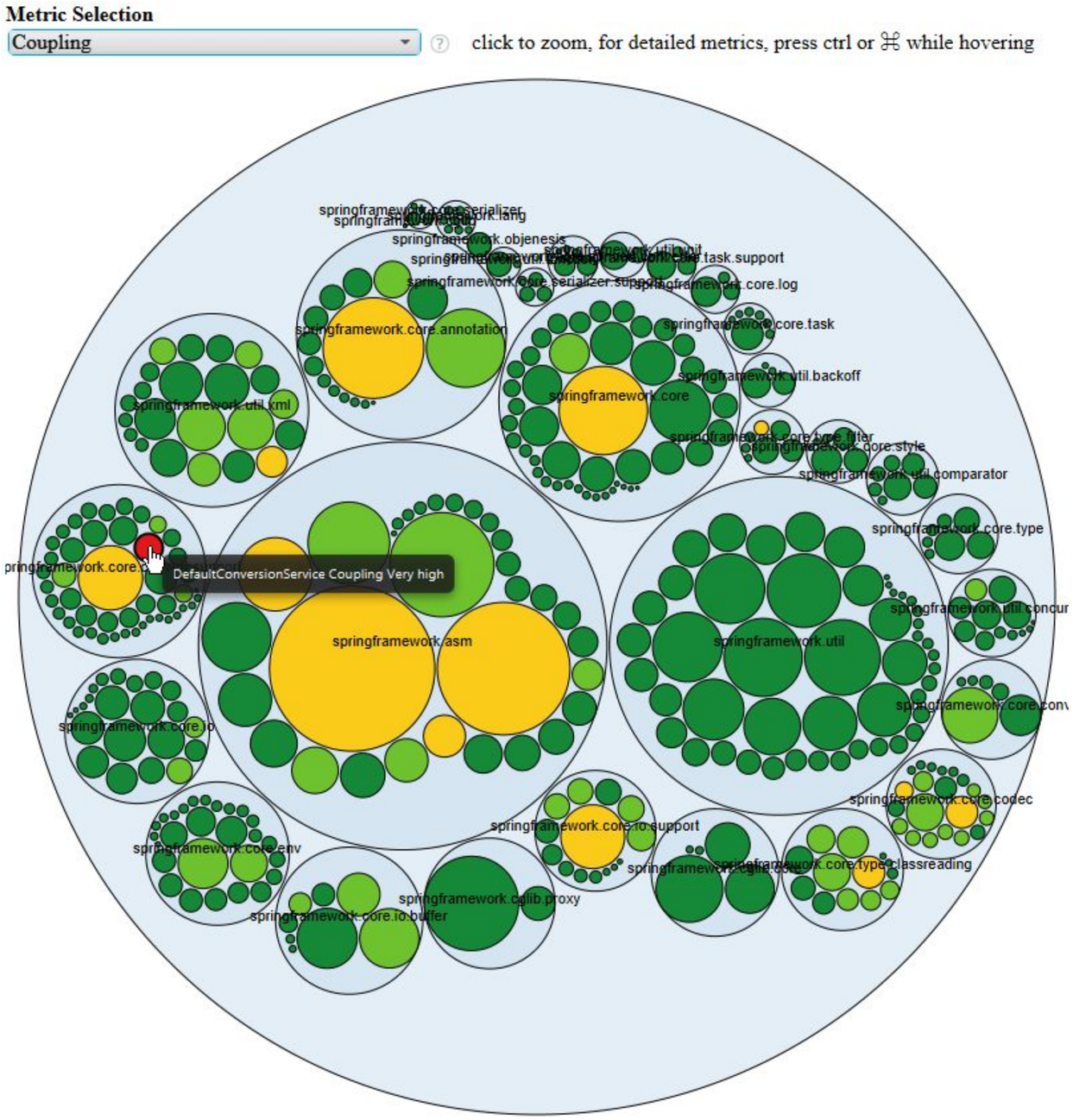
Spring


PathMatchingResourcePatternResolver
TypeDescriptor
ObjectUtils
DefaultConversionService
ClassUtils
AntPathMatcher
GenericConversionService
ReflectionUtils
StringUtils
SynthesizedAnnotationInvocationHandler
AnnotationAwareOrderComparator
AnnotationUtils
AnnotatedElementUtils
MethodMetadataReadingVisitor
ResolvableType
Conventions
MethodParameter
StandardMethodMetadata
StandardAnnotationMetadata
ClassWriter
ClassReader
Frame

Spring


StandardAnnotationMetadata
AnnotationMetadataReadingVisitor
StandardMethodMetadata
MethodWriter
MethodMetadataReadingVisitor
ClassWriter
AnnotationAwareOrderComparator
SymbolTable
LocalVariableTableParameterNameDiscoverer
ResolvableType
MethodParameter
Frame
GenericConversionService
DefaultConversionService
AntPathMatcher
PathMatchingResourcePatternResolver
TypeDescriptor

Spring


Metric Selection
Coupling
click to zoom, for detailed metrics, press ctrl or ⌘ while hovering
springframework.core.annotation
springframework.core
springframework.util.backoff
springframework.util.xml
springframework.core.type.filter
springframework.core.style
springframework.util.comparator
springframework.core.type
springframework.util.concur
springframework.core.conv
springframework.core.codec
springframework.core.type.classreading
springframework.core.io.support
springframework.cglib.proxy
springframework.core.io.buffer
springframework.core.env
springframework.core.io
springframework.asm
springframework.util
DefaultConversionService Coupling Very high

Thanks!

Any questions?