# SDMetrics

The **S**oftware **D**esign **Metrics** tool for the UML™

Last updated: 18-Feb-2011

## UML Design Quality Measurement

- Calculate design metrics for UML designs, e.g.,
  - Coupling between components
  - Size of packages
  - Complexity of classes
  - Etc.
- Check UML design rules
  - Detect incomplete, incorrect, redundant, or inconsistent design
  - Find style problems such as circular dependencies, violation of naming conventions, and more

## Design Properties: Coupling

- The degree to which an element in the design is connected to other elements.
- Distinction: import coupling and export coupling
  - Aka afferent/efferent coupling, fan-in/fan-out
- Impact of high coupling on system quality
  - Decreased maintainability
  - Decreased testability
  - Decreased reliability
  - Decreased reusability

# Example Coupling Metrics

- Class coupling metrics
  - Outgoing/incoming associations or dependencies
  - Class used as parameter type of an operation
  - Messages sent/received by instances of the class
- Package coupling metrics
  - Associations or dependencies from or to classes/interfaces outside the package
- Component coupling metrics
  - Required interfaces, provided interfaces

# Design Properties: Size

- Size of a design element: the number or size of elements it contains.
  - Number of attributes, operations in a class
  - Number of classes, interfaces, in a package
  - Number of states in a state machine
  - ...
- Good indicators of effort
- Large elements indicate poor design
  - E.g. “God” or “Blob” classes

## Design Properties: Complexity

- The connectivity between the elements of a design unit (“count of the edges in a graph”)
  - Number of method calls between lifelines in a sequence/communication diagram
  - Number of transitions in a state chart
  - Number of object/control flows in an activity diagram
- Impact of high complexity on system quality
  - Decreased understandability, testability
  - Increased fault-proneness

## Further Examples of Metrics

- Class inheritance metrics
  - Depth of the class in the inheritance hierarchy
  - Number of children or descendent classes
- The “Martin Metrics” for packages
  - Abstraction (A), instability (I), distance from main sequence (D)
- Metrics for use cases
  - Included or extended use cases
  - Size of a use case in terms of the size of its sequence/activity diagrams

# Design Rule Checking (1)

- Completeness
  - Unused or unreachable elements
  - Underspecified elements: unnamed, missing type,...
- Correctness
  - Well-formedness rules of the UML
  - E.g., circular inheritance, fork state does not target orthogonal states, ...

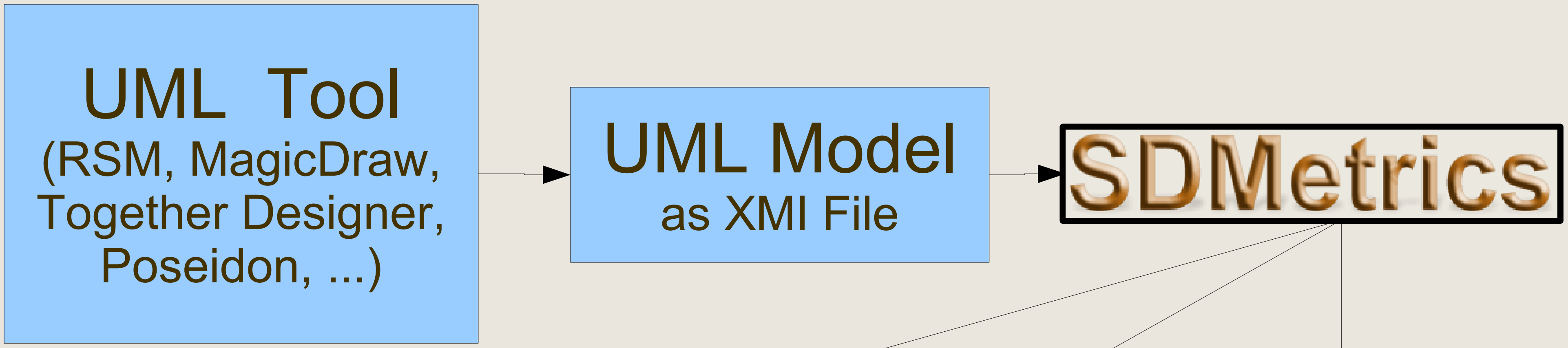
## Design Rule Checking (2)

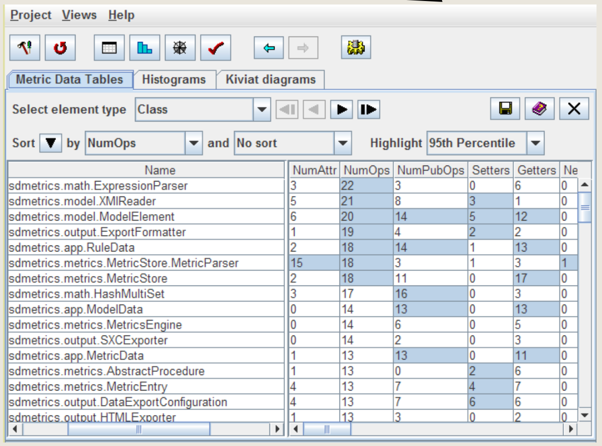
- Style
  - Circular dependencies between classes/packages
  - Controversial design practices, e.g.:
    - Multiple inheritance
    - Use of association classes, n-ary associations
  - Big classes, long parameter lists
- Naming
  - Conventions for capitalization, prefixes, ...
  - Use of programming language keywords, ...

## Benefits of Design Measurement

- Identify potential design problems early on
- Better focus of review and testing efforts
- Increase system quality and quality assurance effectiveness
- Refine your LOC or effort estimates for implementation, testing, maintenance

# From Model to Metrics with XMI (1)

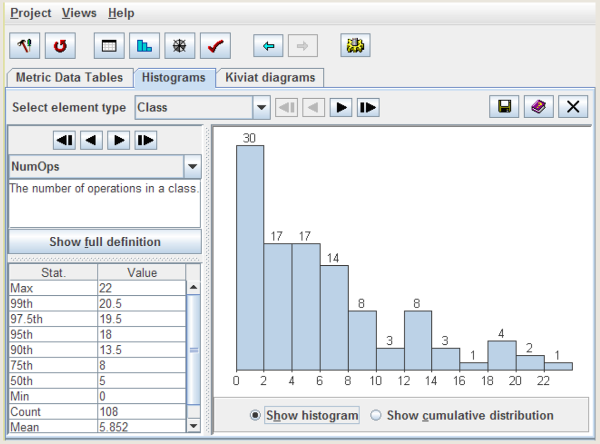




Select element type: Class

Sort ▼ by NumOps and No sort  Highlight 95th Percentile

| Name | NumAttr | NumOps | NumPubOps | Setters | Getters | Ne |
|---|---|---|---|---|---|---|
| sdmetrics.math.ExpressionParser | 3 | 22 | 3 | 0 | 6 | 0 |
| sdmetrics.model.XMIReader | 5 | 21 | 8 | 3 | 1 | 0 |
| sdmetrics.model.ModelElement | 6 | 20 | 14 | 5 | 12 | 0 |
| sdmetrics.output.ExportFormatter | 1 | 19 | 4 | 2 | 2 | 0 |
| sdmetrics.app.RuleData | 2 | 18 | 14 | 1 | 13 | 0 |
| sdmetrics.metrics.MetricStore.MetricParser | 15 | 18 | 3 | 1 | 3 | 1 |
| sdmetrics.metrics.MetricStore | 2 | 18 | 11 | 0 | 17 | 0 |
| sdmetrics.math.HashMultiSet | 3 | 17 | 16 | 0 | 3 | 0 |
| sdmetrics.app.ModelData | 0 | 14 | 13 | 0 | 13 | 0 |
| sdmetrics.metrics.MetricsEngine | 0 | 14 | 6 | 0 | 5 | 0 |
| sdmetrics.output.SXCExporter | 0 | 14 | 2 | 0 | 3 | 0 |
| sdmetrics.app.MetricData | 1 | 13 | 13 | 0 | 11 | 0 |
| sdmetrics.metrics.AbstractProcedure | 1 | 13 | 0 | 2 | 6 | 0 |
| sdmetrics.metrics.MetricEntry | 4 | 13 | 7 | 4 | 7 | 0 |
| sdmetrics.output.DataExportConfiguration | 4 | 13 | 7 | 6 | 6 | 0 |
| sdmetrics.output.HTMLExporter | 1 | 13 | 3 | 0 | 2 | 0 |

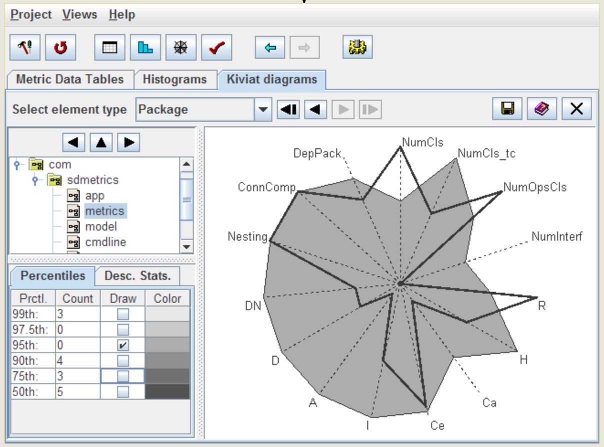


Select element type: Class

NumOps

The number of operations in a class.

| Stat. | Value |
|---|---|
| Max | 22 |
| 99th | 20.5 |
| 97.5th | 19.5 |
| 95th | 18 |
| 90th | 13.5 |
| 75th | 8 |
| 50th | 5 |
| Min | 0 |
| Count | 108 |
| Mean | 5.852 |



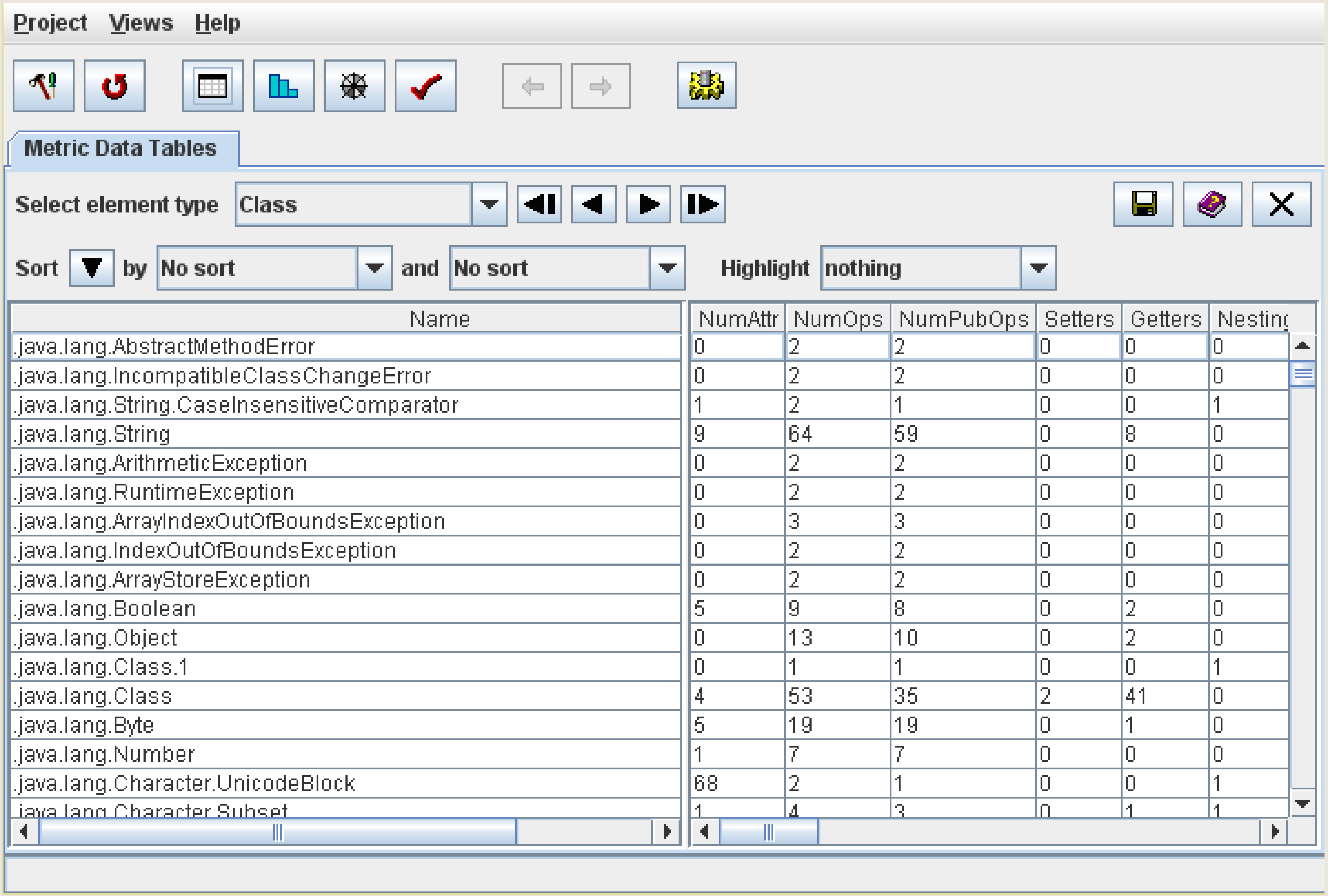
Select element type: Package

| Prctl. | Count | Draw | Color |
|---|---|---|---|
| 99th: | 3 | ☐ | |
| 97.5th: | 0 | ☐ | |
| 95th: | 0 | ✓ | |
| 90th: | 4 | ☐ | |
| 75th: | 3 | ☐ | |
| 50th: | 5 | ☐ | |

# From Model to Metrics with XMI (2)

- XMI: **X**ML **M**etadata **I**nterchange
  - XML-based representation of a UML Model
  - Standardized by OMG; key technology of the MDA
  - Supported by all major UML modeling tools
  - Different versions of XMI: 1.0/1.1/1.2/2.0/2.1
- SDMetrics supports all versions of XMI
  - Works with all modeling tools with XMI export
  - Meta models for UML 1.x and UML 2.x

# Metric Analysis: Table View



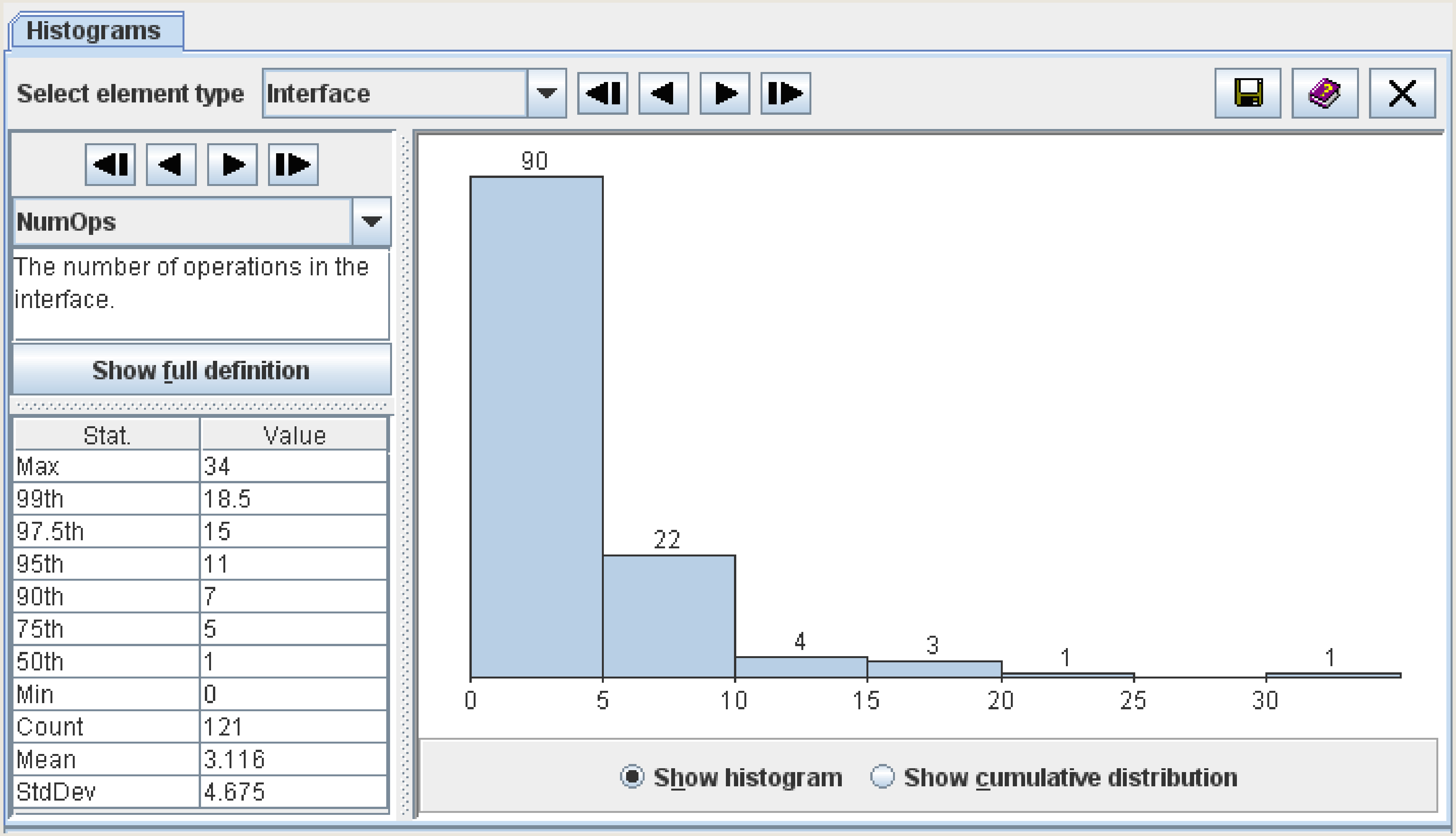
Select element type: Class

Sort ▼ by No sort and No sort

Highlight nothing

| Name | NumAttr | NumOps | NumPubOps | Setters | Getters | Nesting |
|---|---|---|---|---|---|---|
| .java.lang.AbstractMethodError | 0 | 2 | 2 | 0 | 0 | 0 |
| .java.lang.IncompatibleClassChangeError | 0 | 2 | 2 | 0 | 0 | 0 |
| .java.lang.String.CaseInsensitiveComparator | 1 | 2 | 1 | 0 | 0 | 1 |
| .java.lang.String | 9 | 64 | 59 | 0 | 8 | 0 |
| .java.lang.ArithmeticException | 0 | 2 | 2 | 0 | 0 | 0 |
| .java.lang.RuntimeException | 0 | 2 | 2 | 0 | 0 | 0 |
| .java.lang.ArrayIndexOutOfBoundsException | 0 | 3 | 3 | 0 | 0 | 0 |
| .java.lang.IndexOutOfBoundsException | 0 | 2 | 2 | 0 | 0 | 0 |
| .java.lang.ArrayStoreException | 0 | 2 | 2 | 0 | 0 | 0 |
| .java.lang.Boolean | 5 | 9 | 8 | 0 | 2 | 0 |
| .java.lang.Object | 0 | 13 | 10 | 0 | 2 | 0 |
| .java.lang.Class.1 | 0 | 1 | 1 | 0 | 0 | 1 |
| .java.lang.Class | 4 | 53 | 35 | 2 | 41 | 0 |
| .java.lang.Byte | 5 | 19 | 19 | 0 | 1 | 0 |
| .java.lang.Number | 1 | 7 | 7 | 0 | 0 | 0 |
| .java.lang.Character.UnicodeBlock | 68 | 2 | 1 | 0 | 0 | 1 |
| .java.lang.Character.Subset | 1 | 4 | 3 | 0 | 1 | 1 |

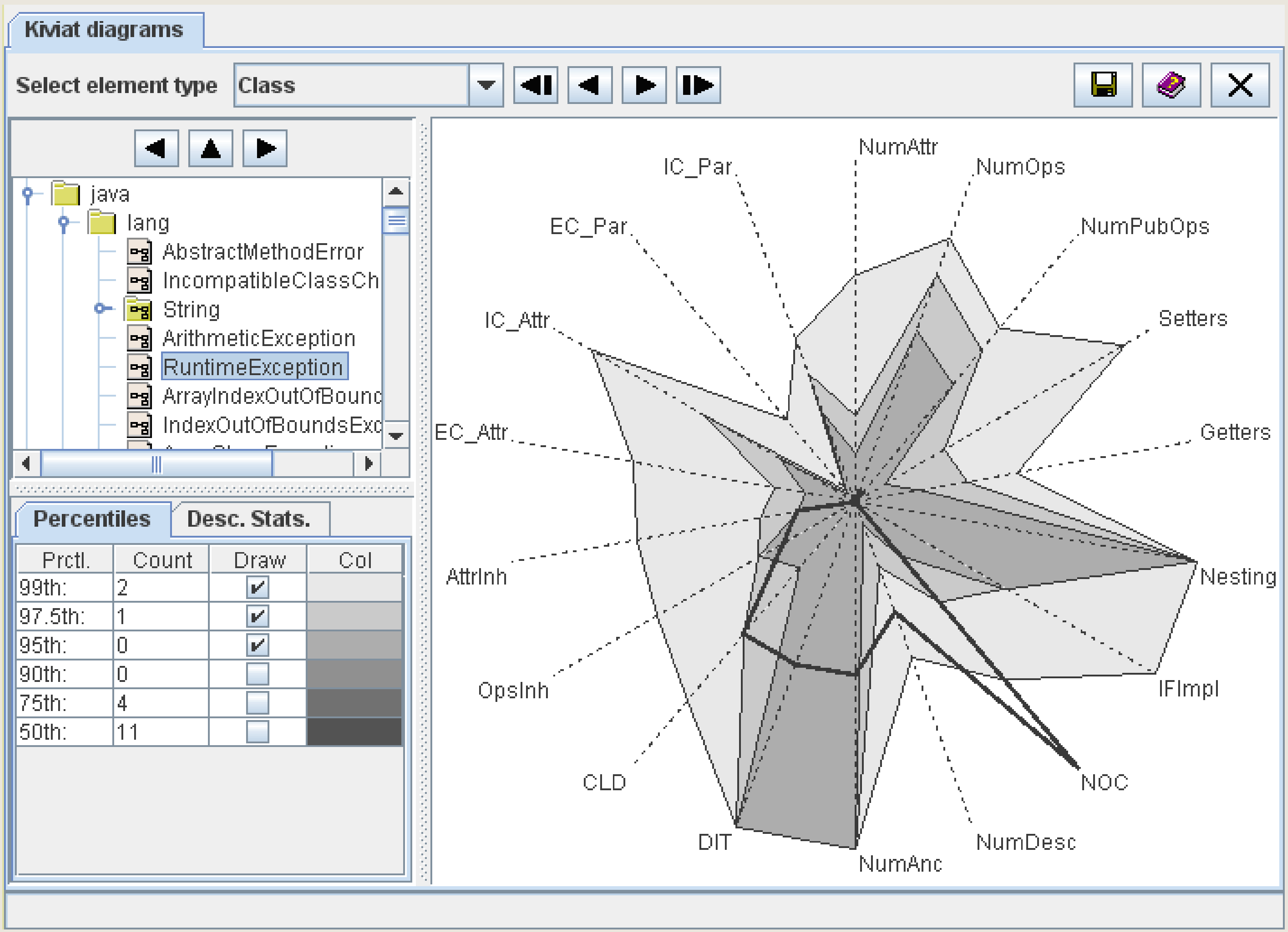
# Metric Analysis: Metric View



The number of operations in the interface.

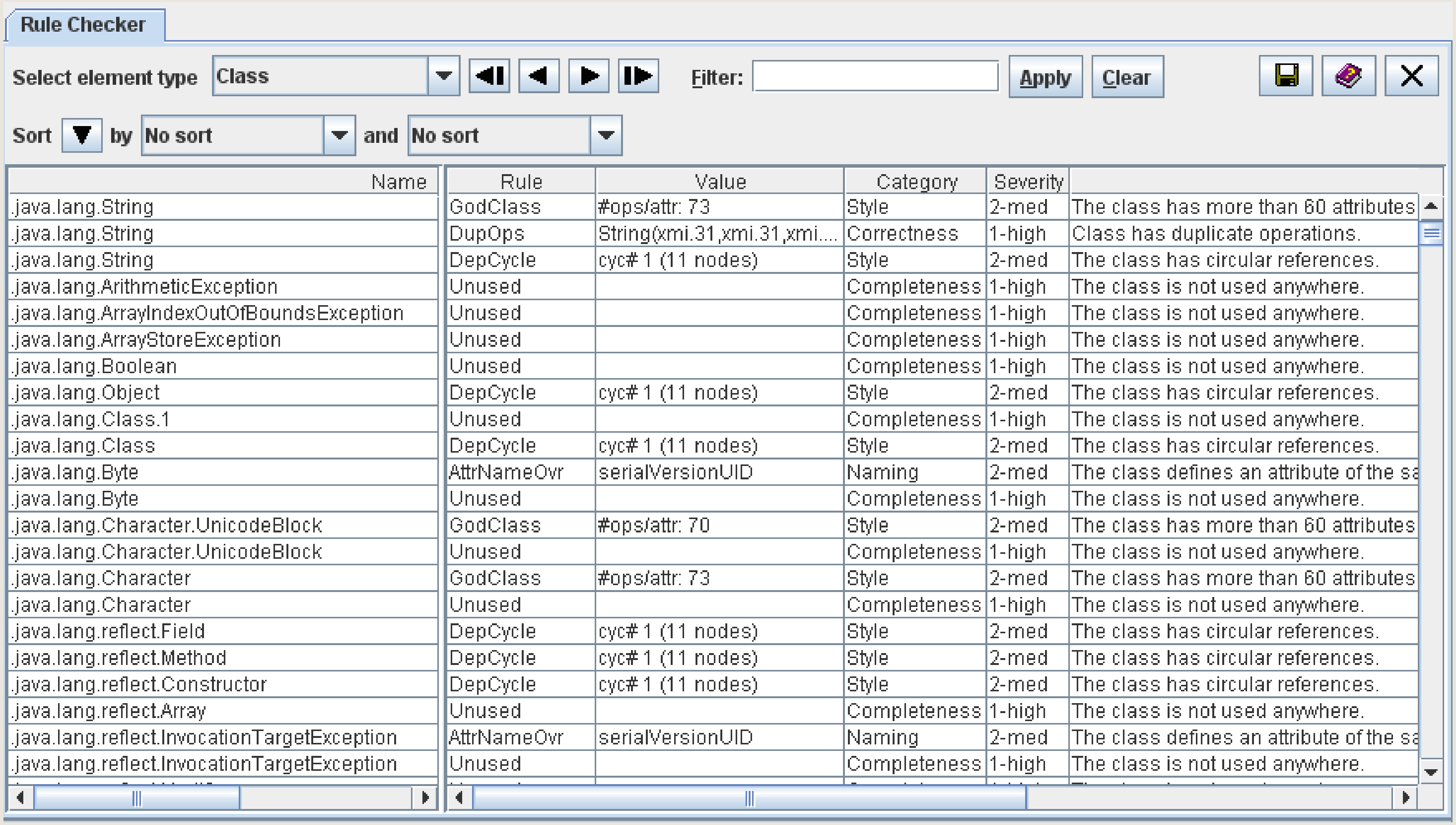
| Stat. | Value |
|---|---|
| Max | 34 |
| 99th | 18.5 |
| 97.5th | 15 |
| 95th | 11 |
| 90th | 7 |
| 75th | 5 |
| 50th | 1 |
| Min | 0 |
| Count | 121 |
| Mean | 3.116 |
| StdDev | 4.675 |

# Metric Analysis: Element View



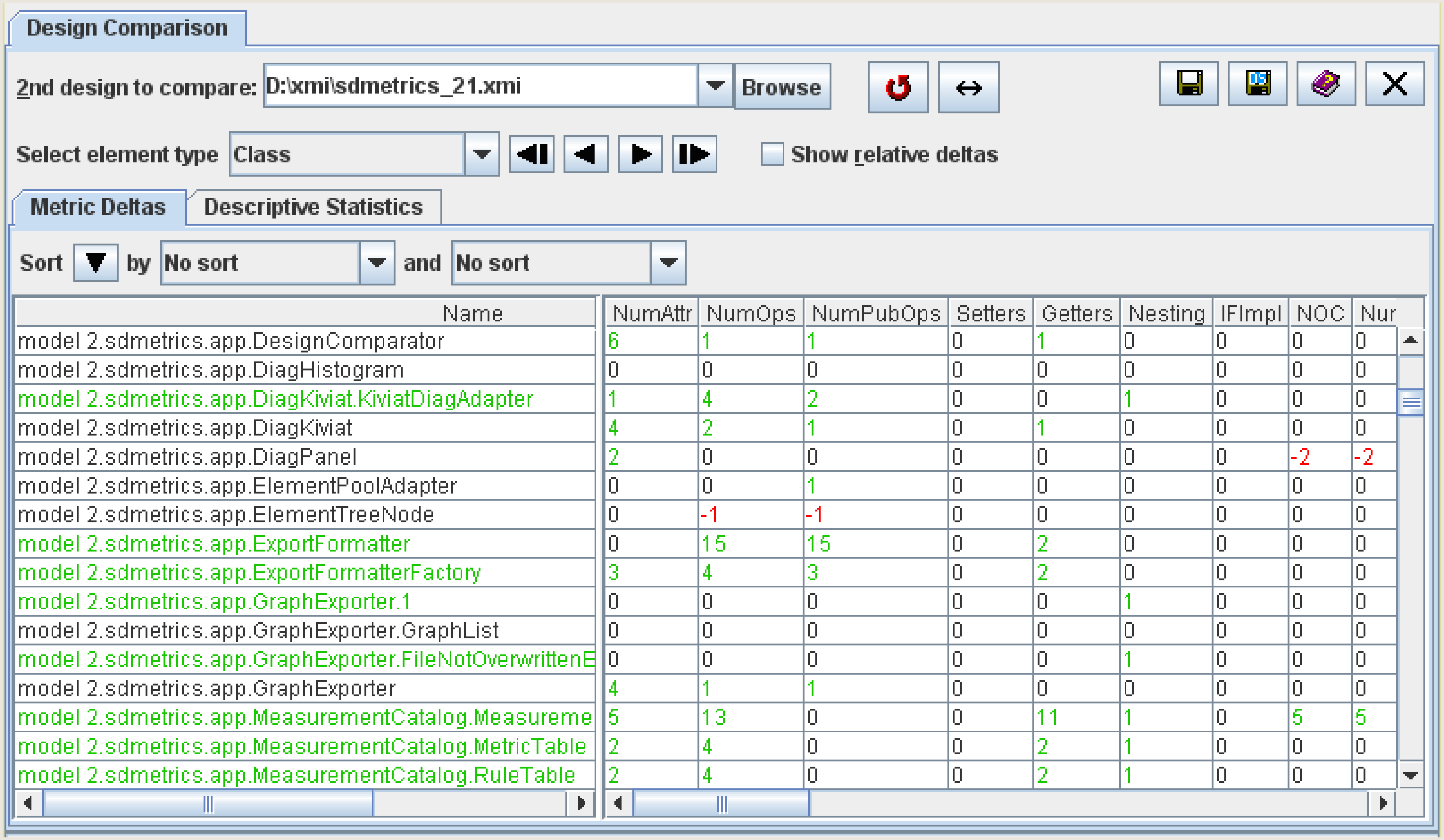
| Prctl. | Count | Draw | Col |
|---|---|---|---|
| 99th: | 2 | ✓ | ▨ |
| 97.5th: | 1 | ✓ | ▨ |
| 95th: | 0 | ✓ | ▨ |
| 90th: | 0 | ☐ | ▨ |
| 75th: | 4 | ☐ | ▨ |
| 50th: | 11 | ☐ | ■ |

# Design Rule Checking



| Name | Rule | Value | Category | Severity | |
|---|---|---|---|---|---|
| .java.lang.String | GodClass | #ops/attr: 73 | Style | 2-med | The class has more than 60 attributes |
| .java.lang.String | DupOps | String(xmi.31,xmi.31,xmi.... | Correctness | 1-high | Class has duplicate operations. |
| .java.lang.String | DepCycle | cyc# 1 (11 nodes) | Style | 2-med | The class has circular references. |
| .java.lang.ArithmeticException | Unused | | Completeness | 1-high | The class is not used anywhere. |
| .java.lang.ArrayIndexOutOfBoundsException | Unused | | Completeness | 1-high | The class is not used anywhere. |
| .java.lang.ArrayStoreException | Unused | | Completeness | 1-high | The class is not used anywhere. |
| .java.lang.Boolean | Unused | | Completeness | 1-high | The class is not used anywhere. |
| .java.lang.Object | DepCycle | cyc# 1 (11 nodes) | Style | 2-med | The class has circular references. |
| .java.lang.Class.1 | Unused | | Completeness | 1-high | The class is not used anywhere. |
| .java.lang.Class | DepCycle | cyc# 1 (11 nodes) | Style | 2-med | The class has circular references. |
| .java.lang.Byte | AttrNameOvr | serialVersionUID | Naming | 2-med | The class defines an attribute of the sa |
| .java.lang.Byte | Unused | | Completeness | 1-high | The class is not used anywhere. |
| .java.lang.Character.UnicodeBlock | GodClass | #ops/attr: 70 | Style | 2-med | The class has more than 60 attributes |
| .java.lang.Character.UnicodeBlock | Unused | | Completeness | 1-high | The class is not used anywhere. |
| .java.lang.Character | GodClass | #ops/attr: 73 | Style | 2-med | The class has more than 60 attributes |
| .java.lang.Character | Unused | | Completeness | 1-high | The class is not used anywhere. |
| .java.lang.reflect.Field | DepCycle | cyc# 1 (11 nodes) | Style | 2-med | The class has circular references. |
| .java.lang.reflect.Method | DepCycle | cyc# 1 (11 nodes) | Style | 2-med | The class has circular references. |
| .java.lang.reflect.Constructor | DepCycle | cyc# 1 (11 nodes) | Style | 2-med | The class has circular references. |
| .java.lang.reflect.Array | Unused | | Completeness | 1-high | The class is not used anywhere. |
| .java.lang.reflect.InvocationTargetException | AttrNameOvr | serialVersionUID | Naming | 2-med | The class defines an attribute of the sa |
| .java.lang.reflect.InvocationTargetException | Unused | | Completeness | 1-high | The class is not used anywhere. |

# Design Comparisons



Design Comparison

2nd design to compare: D:\xmi\sdmetrics_21.xmi Browse

Select element type Class

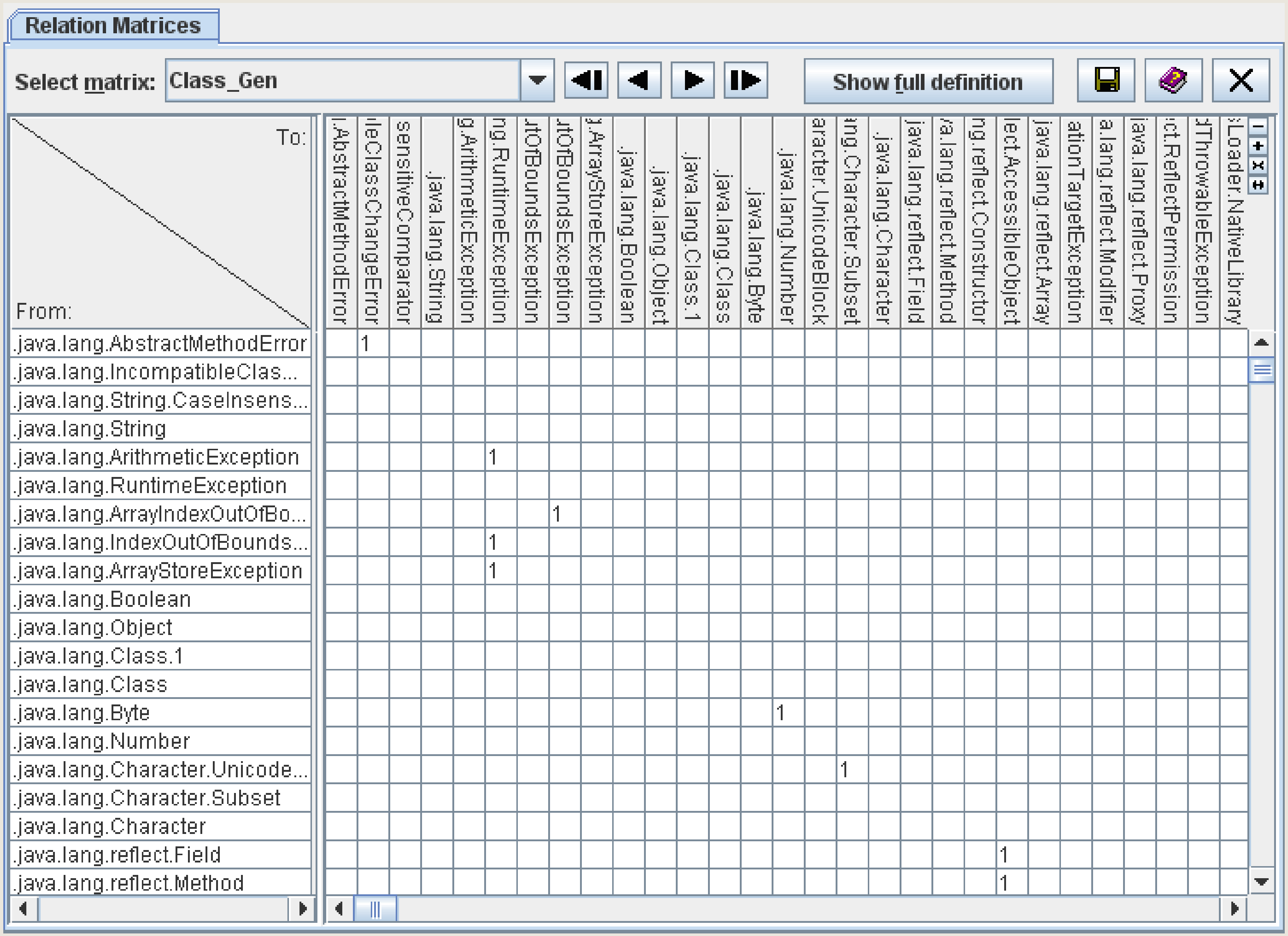
Show relative deltas

Metric Deltas | Descriptive Statistics

Sort by No sort and No sort

| Name | NumAttr | NumOps | NumPubOps | Setters | Getters | Nesting | IFImpl | NOC | Nur |
|---|---|---|---|---|---|---|---|---|---|
| model 2.sdmetrics.app.DesignComparator | 6 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| model 2.sdmetrics.app.DiagHistogram | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| model 2.sdmetrics.app.DiagKiviat.KiviatDiagAdapter | 1 | 4 | 2 | 0 | 0 | 1 | 0 | 0 | 0 |
| model 2.sdmetrics.app.DiagKiviat | 4 | 2 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| model 2.sdmetrics.app.DiagPanel | 2 | 0 | 0 | 0 | 0 | 0 | 0 | -2 | -2 |
| model 2.sdmetrics.app.ElementPoolAdapter | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| model 2.sdmetrics.app.ElementTreeNode | 0 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 |
| model 2.sdmetrics.app.ExportFormatter | 0 | 15 | 15 | 0 | 2 | 0 | 0 | 0 | 0 |
| model 2.sdmetrics.app.ExportFormatterFactory | 3 | 4 | 3 | 0 | 2 | 0 | 0 | 0 | 0 |
| model 2.sdmetrics.app.GraphExporter.1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| model 2.sdmetrics.app.GraphExporter.GraphList | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| model 2.sdmetrics.app.GraphExporter.FileNotOverwrittenE | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| model 2.sdmetrics.app.GraphExporter | 4 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| model 2.sdmetrics.app.MeasurementCatalog.Measureme | 5 | 13 | 0 | 0 | 11 | 1 | 0 | 5 | 5 |
| model 2.sdmetrics.app.MeasurementCatalog.MetricTable | 2 | 4 | 0 | 0 | 2 | 1 | 0 | 0 | 0 |
| model 2.sdmetrics.app.MeasurementCatalog.RuleTable | 2 | 4 | 0 | 0 | 2 | 1 | 0 | 0 | 0 |

# Relation Matrices



| To: From: | l.AbstractMethodError | leClassChangeError | sensitiveComparator | .java.lang.String | g.ArithmeticException | ng.RuntimeException | utOfBoundsException | utOfBoundsException | j.ArrayStoreException | .java.lang.Boolean | .java.lang.Object | .java.lang.Class.1 | .java.lang.Class | .java.lang.Byte | .java.lang.Number | aracter.UnicodeBlock | ing.Character.Subset | .java.lang.Character | java.lang.reflect.Field | /a.lang.reflect.Method | ng.reflect.Constructor | lect.AccessibleObject | java.lang.reflect.Array | ationTargetException | a.lang.reflect.Modifier | java.lang.reflect.Proxy | ct.ReflectPermission | dThrowableException | sLoader.NativeLibrary |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| .java.lang.AbstractMethodError | | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.IncompatibleClas... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.String.CaseInsens... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.String | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.ArithmeticException | | | | | | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.RuntimeException | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.ArrayIndexOutOfBo... | | | | | | | | 1 | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.IndexOutOfBounds... | | | | | | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.ArrayStoreException | | | | | | 1 | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.Boolean | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.Object | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.Class.1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.Class | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.Byte | | | | | | | | | | | | | | | 1 | | | | | | | | | | | | | | |
| .java.lang.Number | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.Character.Unicode... | | | | | | | | | | | | | | | | | 1 | | | | | | | | | | | | |
| .java.lang.Character.Subset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.Character | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| .java.lang.reflect.Field | | | | | | | | | | | | | | | | | | | | | | 1 | | | | | | | |
| .java.lang.reflect.Method | | | | | | | | | | | | | | | | | | | | | | 1 | | | | | | | |

# Data Export

- Detailed metric data analysis with statistical software packages or spreadsheets
- Export of data tables
  - Tab or comma-separated text files
  - HTML
  - XML for Microsoft Excel
  - OpenOffice.org Calc
- Export of graphs
  - SVG, PNG, JPG

# Summary of SDMetrics' Features

- Over 120 design metrics, 130 design rules
  - Cover all diagram types of the UML
  - Users can define new metrics and rules
- Works with all UML tools with XMI export
  - Customizable XMI import
- Batch processing via command line interface
- Fast execution
  - Analyses large designs with hundreds of thousands of model elements within seconds

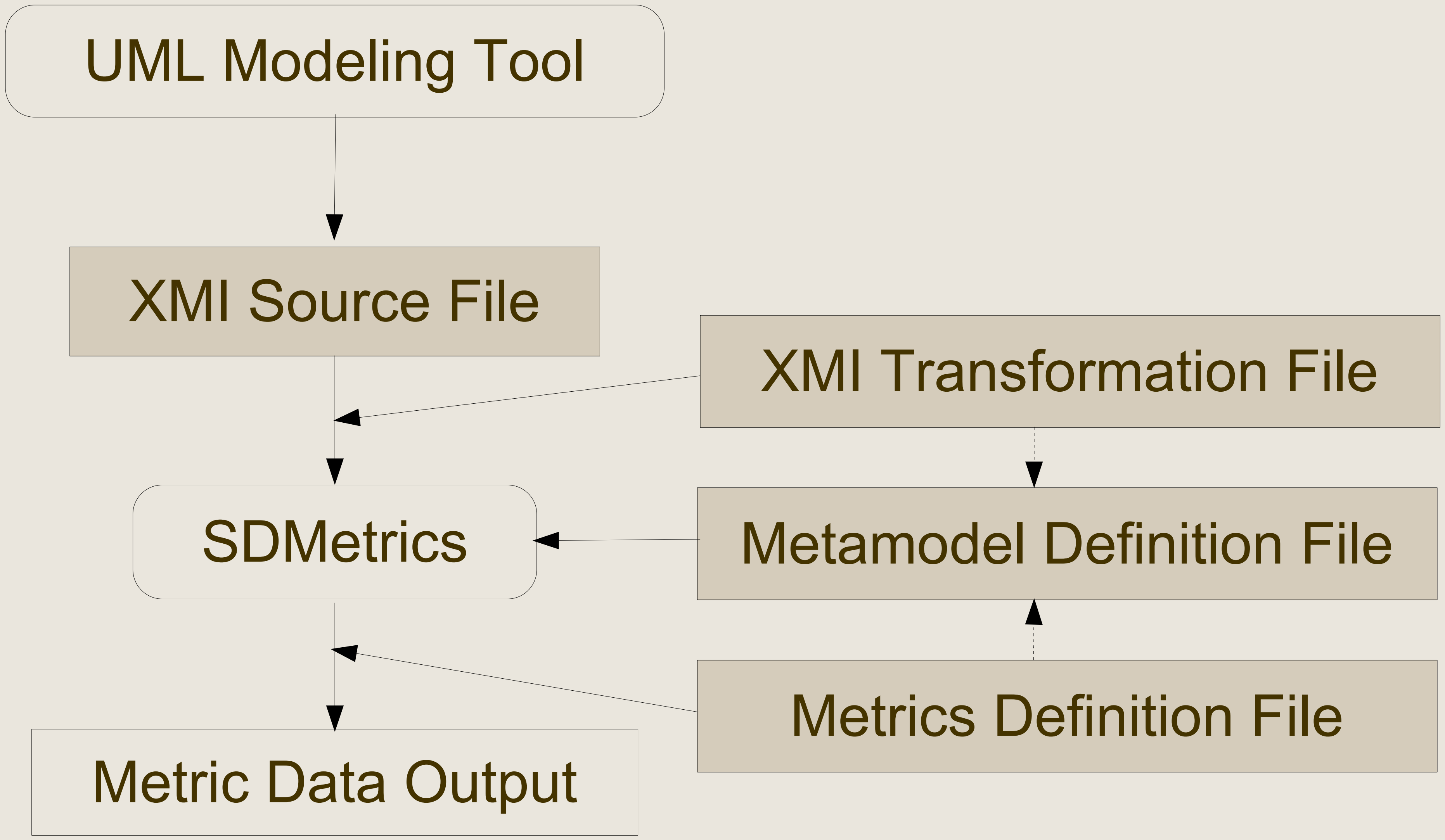
## Customizing SDMetrics

User-defined UML meta models

User-defined XMI import

User-defined metrics and rules

# SDMetrics Project Files



## Meta Model Definition File

- Defines the model elements that SDMetrics knows:
  - Meta classes
  - Attributes
  - Relationships
- Defined as far as is needed for metric calculation and design rule checking
  - Not a complete 1:1 representation of the standard UML meta models
  - Default meta models for UML1.3/1.4 and UML2.x

## Meta Model Definition File Sample

```
<modelelement name="class">
  <attribute name="visibility"
   type=”data” multiplicity=”one”/>
  <attribute name="abstract" type=”data”/>
  <attribute name="ownedattributes"
   type="ref" multiplicity="many" />
  <attribute name="ownedoperations"
   type="ref" multiplicity="many" />
  ...
</modelelement>
```

# XMI Transformation File (1)

- How to retrieve information of the model elements and attributes from XMI files
- Defines a mapping:

UML meta model as manifested in the XMI file

$\Updownarrow$

SDMetrics' simplified meta model

# Excerpt from an XMI File

```
<ownedMember xmi:type='UML:Class'
  xmi:id='id345' visibility='public'
  isAbstract='false' name='MyClass'>
  <ownedAttribute xmi:id='id1138'
   name=”myAttr” visibility='private'
   type='id42'/>
  <ownedOperation xmi:id='id1139'
   name='myOperation' visibility='public'>
   <ownedParameter xmi:id='1140'
    name='par1' type='id123' />
  </ownedOperation>
  ...
</ownedMember>
```

# XMI Transformation File Sample

```
<xmitransformation modelelement="class"
  xmipattern="UML:Class" recurse="true">
  <trigger name="visibility"
   type="attrval" attr="visibility"/>
  <trigger name="abstract" type="attrval"
   attr="isAbstract"/>
  <trigger name="ownedattributes"
   type="xmi2assoc" src="UML:Property"
   attr="ownedAttribute" />
  <trigger name="ownedoperations"
   type="xmi2assoc" src="UML:Operation"
   attr="ownedOperation" />
...
```

# XMI Transformation File (2)

- UML model exchange via XMI is difficult in practice
  - Different versions of the XMI
  - Implementations not fully compliant with standards

➔ Customizable XMI import to account for this!

- Default XMI transformation files:
  - For UML1.3/1.4 meta model and XMI 1.x files
  - For UML2.x meta model and XMI 2.0 / 2.1 files
- Plus specialized files for non-conforming tools

# Metric Definition File

- Contains the definitions of:
  - Design metrics
  - Design rules
  - Relation matrices
  - Complete documentation with glossary and literature references

- SDMetrics provides default metric definitions for UML1.x and UML2.x meta models

# Example Metric Definition

```
<metric name="NumPubOps" domain="class"
  category="Size">
  <description>The number of public
   operations in a class.((p))
   Same as metric metric://class/NumOps/,
   but only counts operations with public
   visibility.
   ((ul))((li))Also known as: NPM (Number
   of Public Methods) ref://LK94/.((/ul))
  </description>
  <projection relset="ownedoperations"
  condition="visibility='public'"/>

</metric>
```

## Example Metric Definition (2)

- Definitions of intermediate “helper” sets and metrics
  - E.g. sets “InAssoc” and “OutAssoc” of incoming and outgoing associations for an interface

```
<metric name="Assoc" domain="interface"
  category="Coupling">
  <description>The number of associations
   the interface participates in.
  </description>
  <compoundmetric
   term="size(InAssoc+OutAssoc)" />

</metric>
```

## Example Rule Definition

```
<rule name="OutgoingAssoc" severity="low"
  domain="interface" category="Style" >
  <description>The interface has outgoing
   associations.((p))
   ref://Oes04/ suggests to avoid this.
  </description>
  <violation condition="size(OutAssoc)!=0"
   value="'#Assoc: '+size(OutAssoc)" />

</rule>
```

## Applications of Design Metrics

Building Quality Benchmarks

Building Predictive Models

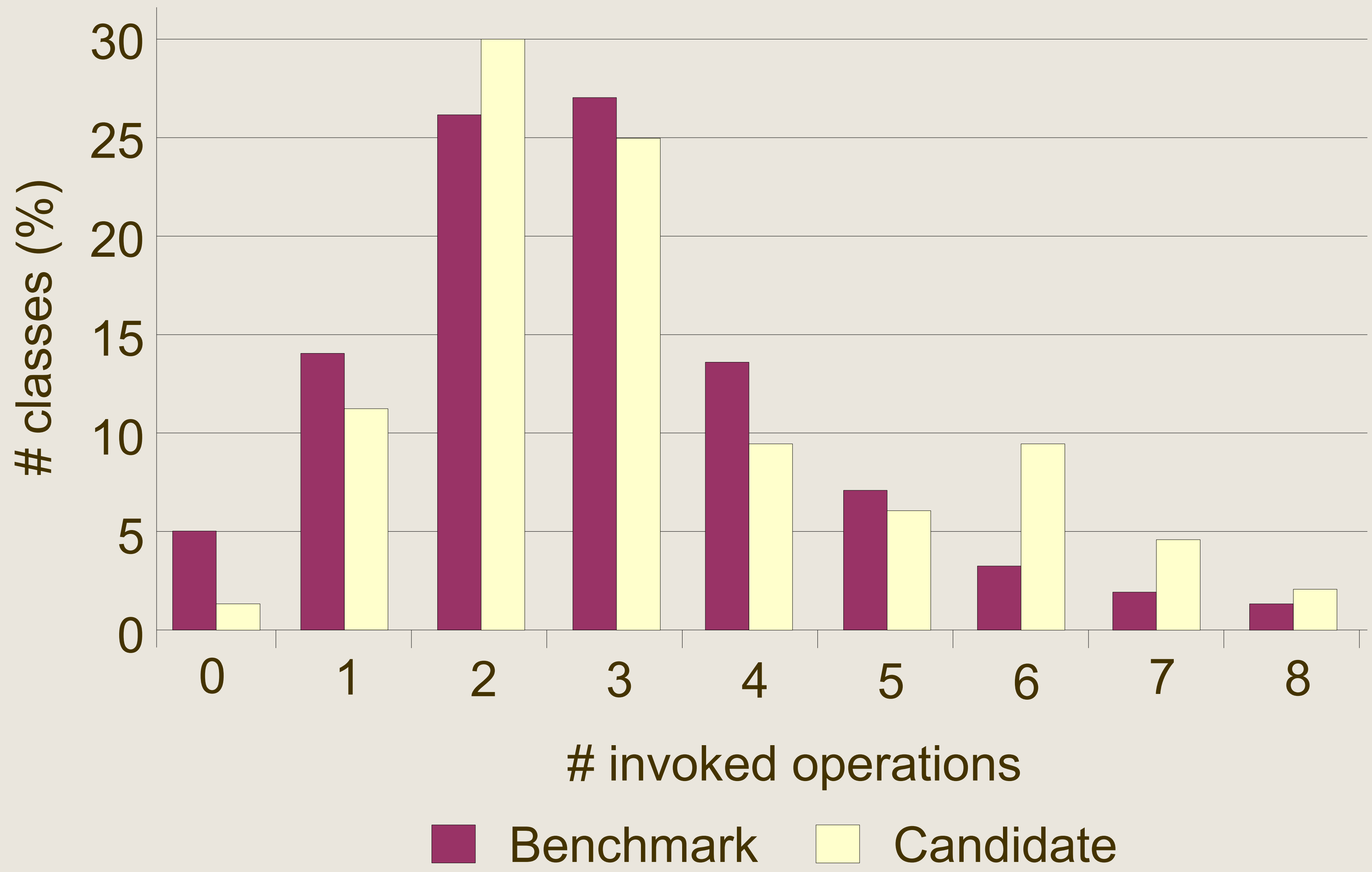
# Quality Benchmarks (1)

- Scenario: acquisition of new components
  - Developed in-house or bought from external source
- Is the design of the components acceptable?
  - Demand rework if there are expected maintainability/reliability problems
- Comprehensive inspection/testing often too expensive
  - Need a pointer to potentially critical design areas to inspect or test

## Quality Benchmarks (2)

- Build a database of measurement values
  - Selected measures of size, coupling, complexity
  - Obtained from existing proven, “tried and tested” components
- Apply measurements to new components
- Compare obtained measurement distributions

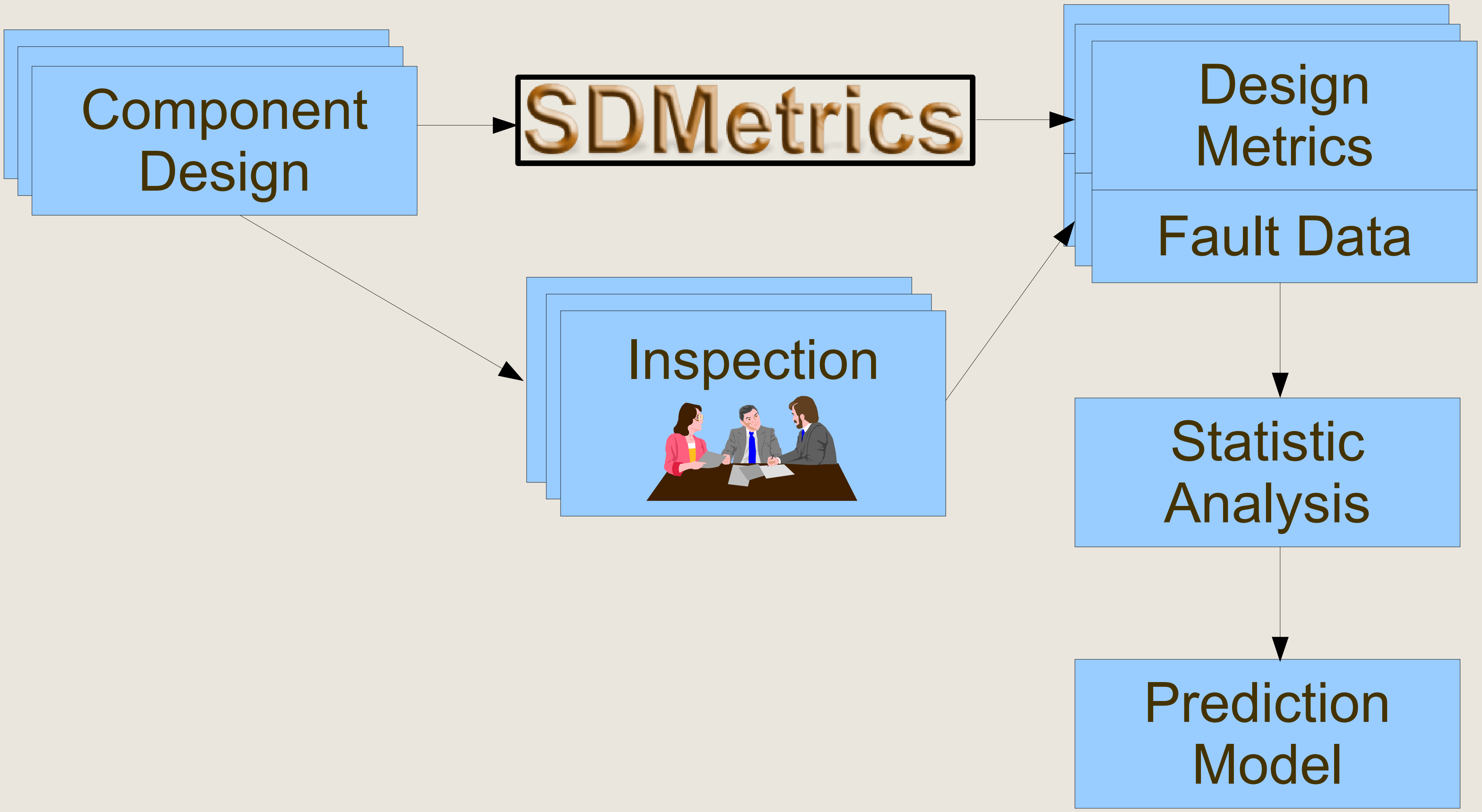
## Quality Benchmarks (3)



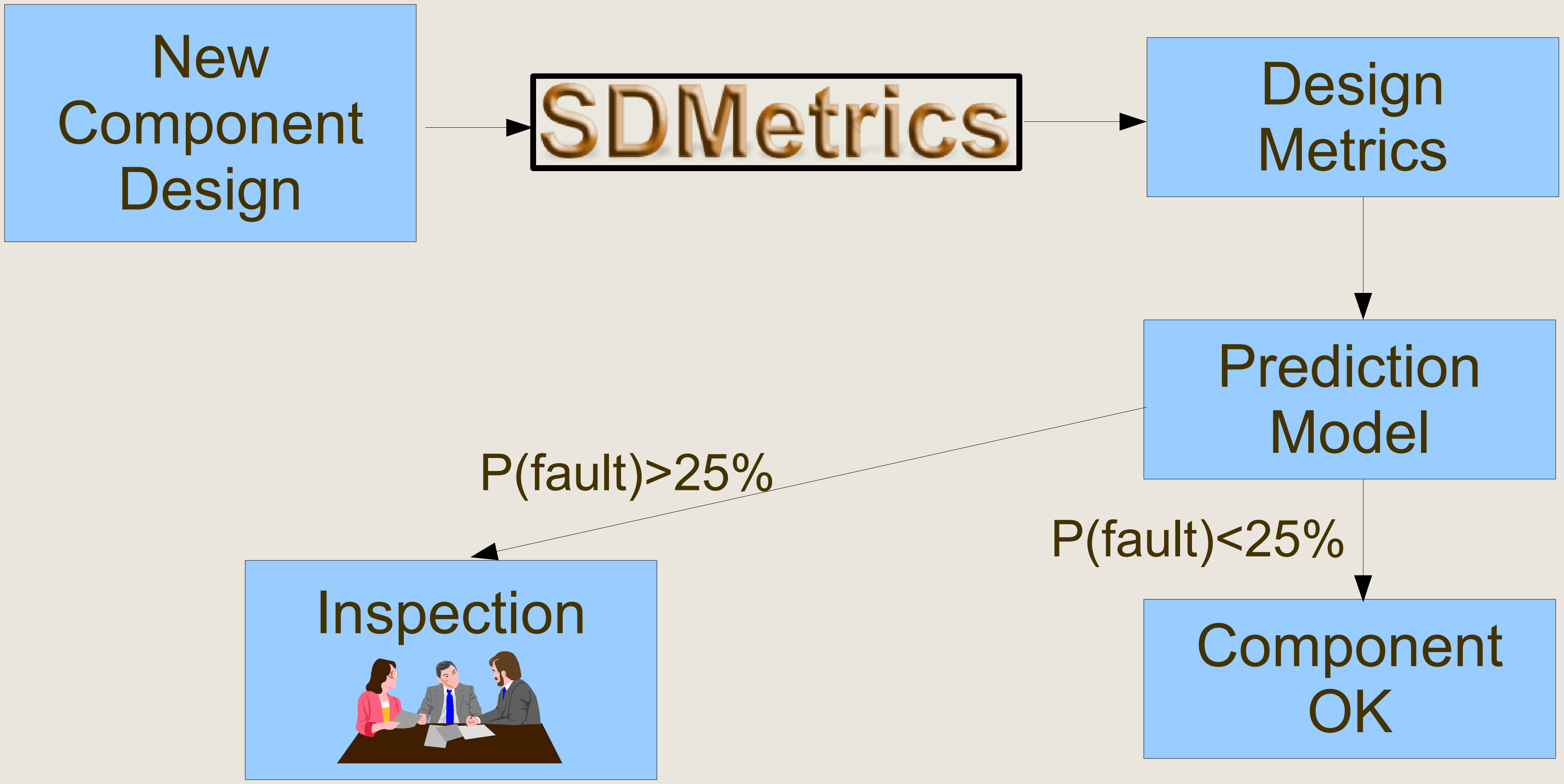
## Prediction Models

- Design metrics data difficult to interpret
  - component XYZ has “coupling=5”, “complexity=7”
- What does this mean in terms of …
  - future maintenance work for this component
  - likelihood that the component still contains defects
- Quantify the relationship between metrics data and system quality for better interpretation

# Building A Prediction Model



# Using the Prediction Model



# Prediction Model Pros and Cons

- Pros
  - Captures the combined effect of multiple influencing factors in one cohesive model
  - Translates non-interpretable internal quality data to easily interpreted external quality data
- Cons
  - Very data intensive
  - Requires statistical expertise