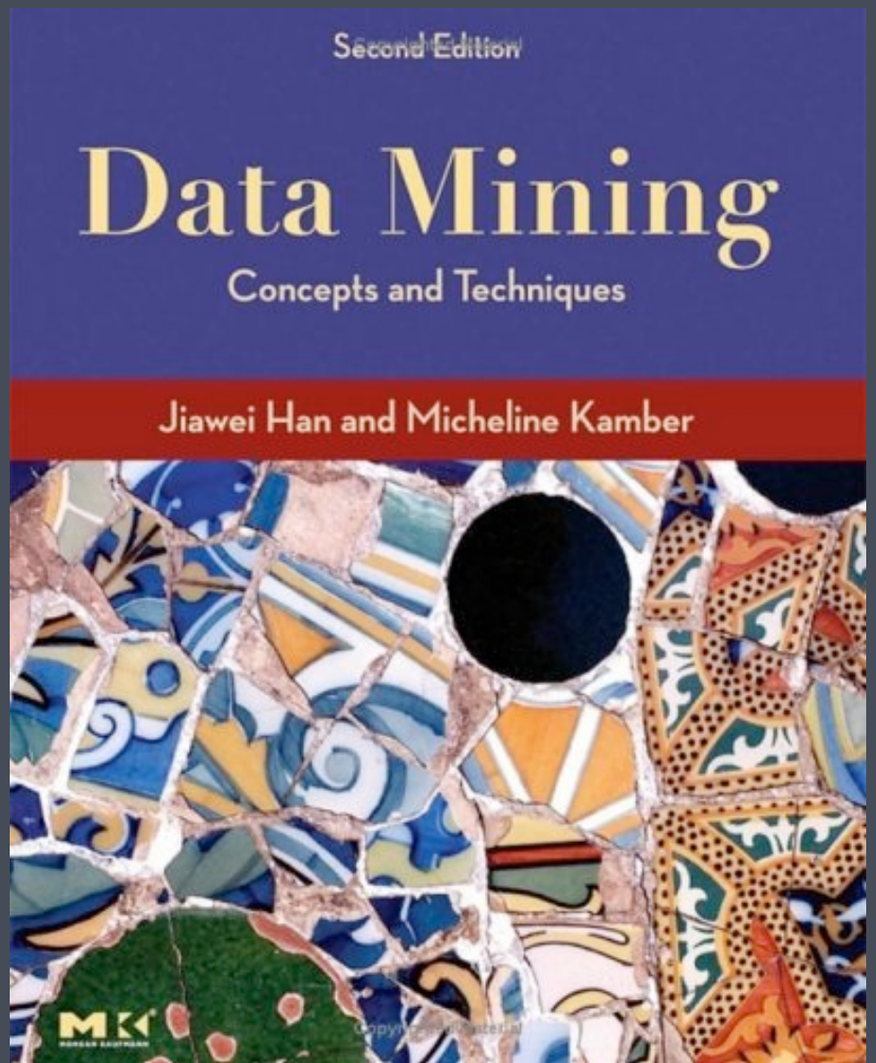
MINING SOFTWARE REPOSITORIES

Software Engineering Course 2009

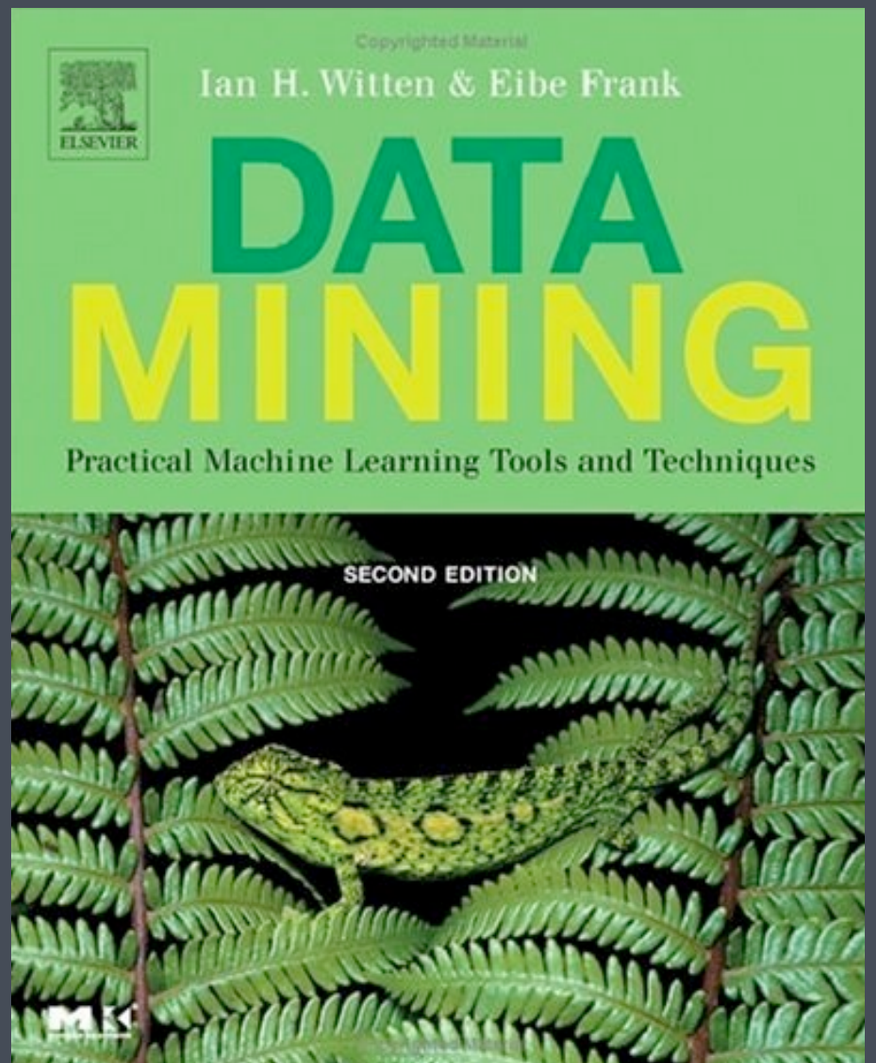
Kim Herzig - Saarland University

Books



Data Mining: Concepts and Techniques

by Jiawei Han & Micheline Kamber



Data Mining: Practical Machine Learning Tools and Techniques

by Ian H. Witten & Eibe Frank

Imagine

You as Quality Manager

Imagine

- 30,000 classes
- ~ 5.5 million lines of code
- ~3000 defect per release
- 700 developers

You as Quality Manager

Your product

Your Boss


Test the system!
You have 6 months, $500,000.
And don't miss any bug!

The Problem

- Not enough **time** to test everything
  - What to test? What to test first?

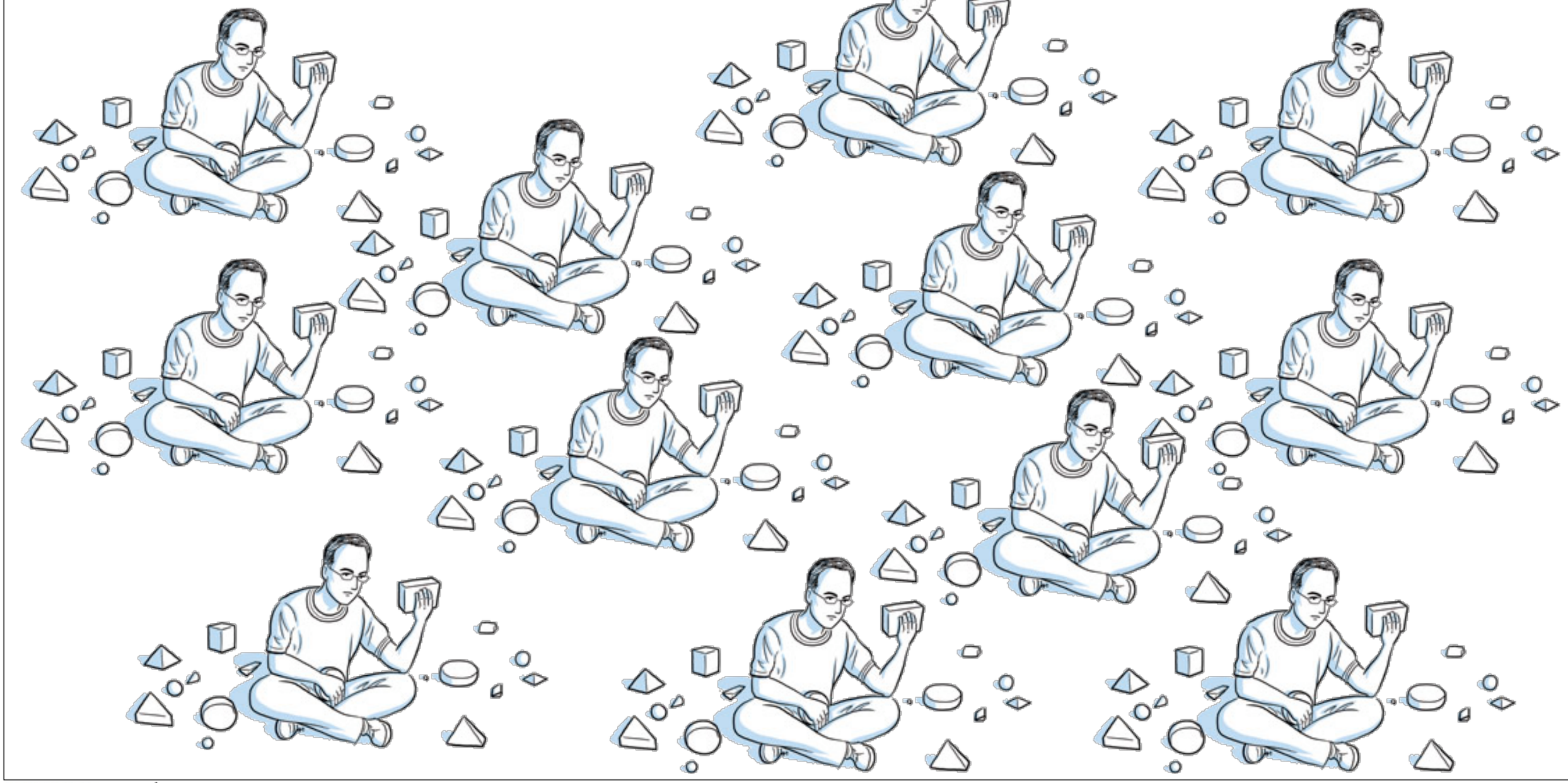
- Not enough **money** to pay enough testers
  - To which extend?

Central question:

Where are the most defect prone entities in my system?

Your Testers

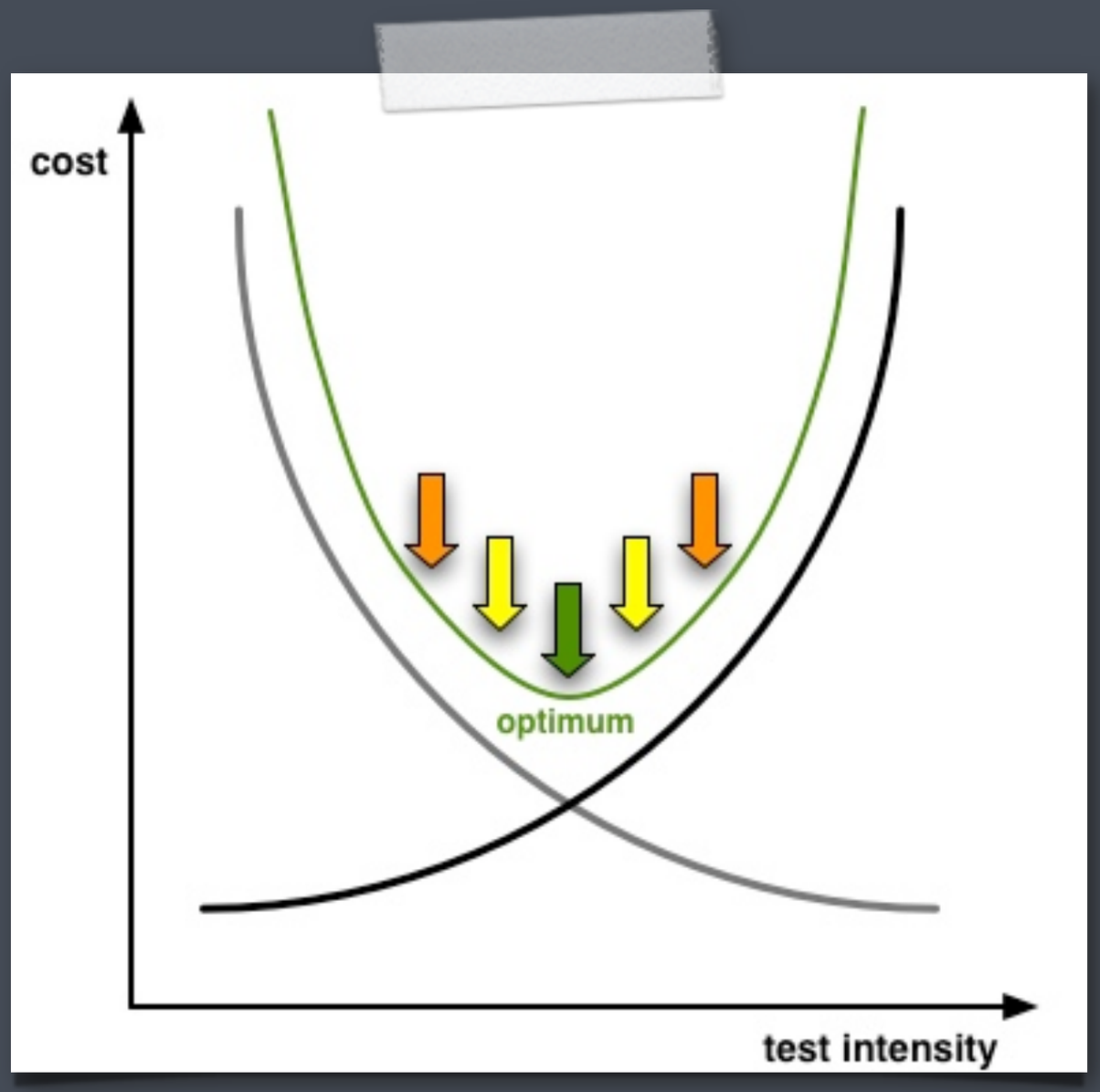
Your Testers



We need efficiency!

We need efficiency!

We need efficiency!



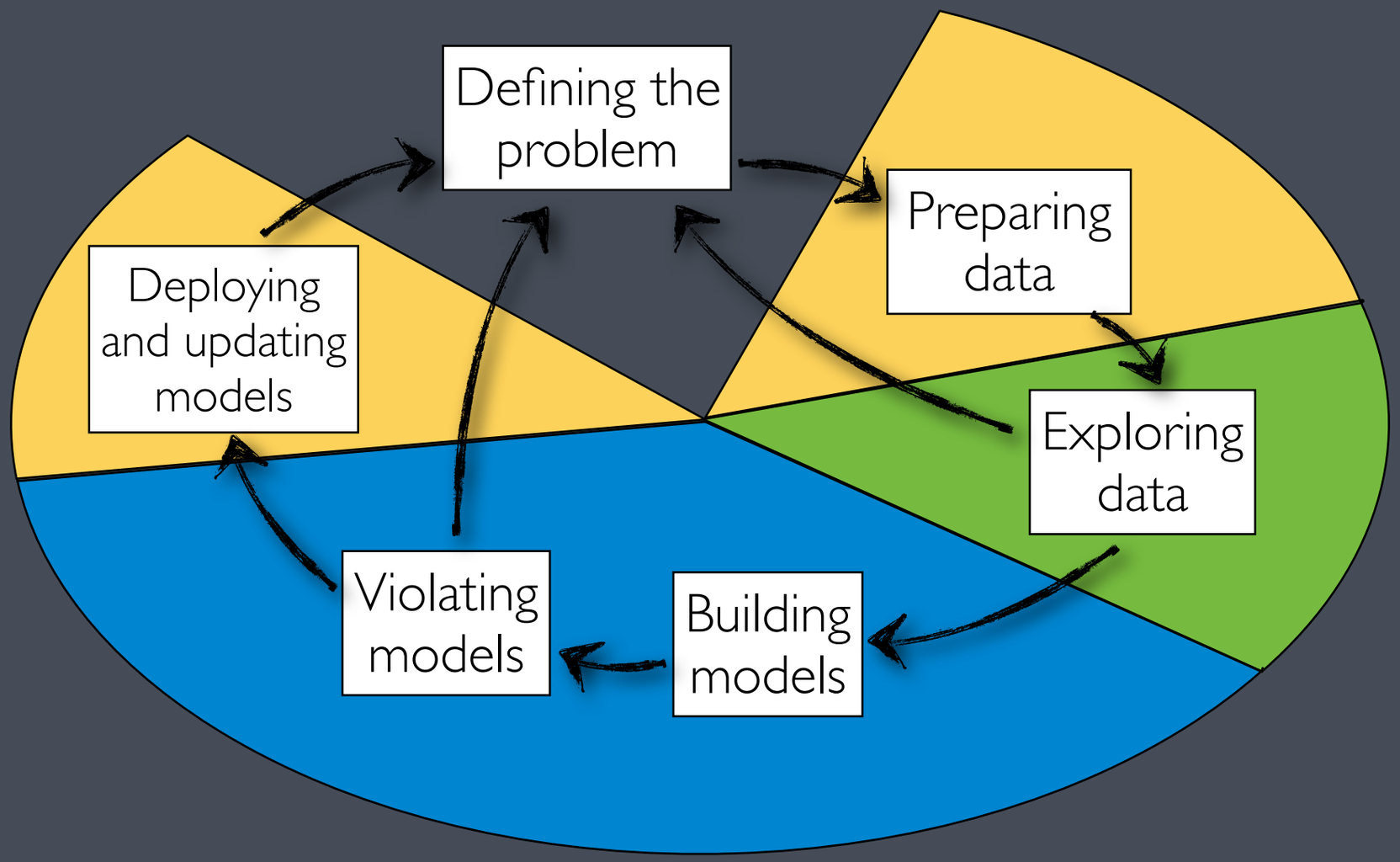
Can we learn from history?
... to predict or estimate the future?

data mining

What is data mining?

Data mining is the process of discovering actionable information from large sets of data.

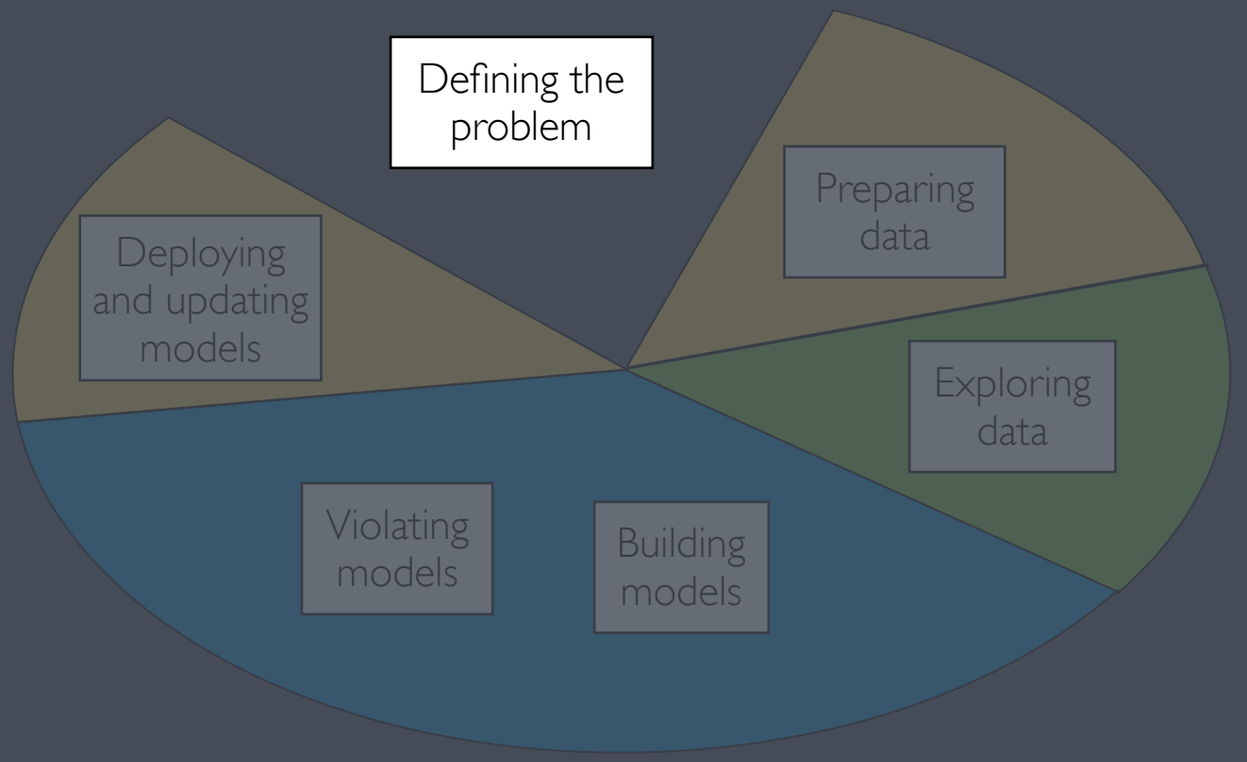
The Mining Model



http://technet.microsoft.com/en-us/library/ms174949.aspx

Step 1: Defining Problem

- Clearly define the problem
  - What are you looking for?
  - Scope of problem
  - Types of relationships
- Define how to evaluate
  - Prediction, recommendation or just patterns



Defect Prediction Problem

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

Which source code entities should we test most?

Defect Prediction Problem

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

Which source code entities should we test most?

Which are the most defect prone entities in my system?

Defect Prediction Problem

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

Which source code entities should we test most?

Which are the most defect prone entities in my system?

In the past, which entities had the most defects?

Defect Prediction Problem

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

Which source code entities should we test most?

Which properties of source code entities correlate with defects?

Which are the most defect prone entities in my system?

In the past, which entities had the most defects?

Data Sources

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

Bug Database

Version Archive

Source Code

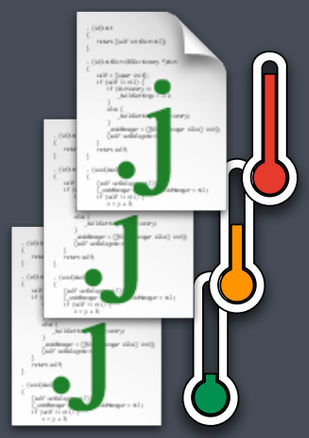
Data Sources

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

Bug Database

Version Archive

Source Code



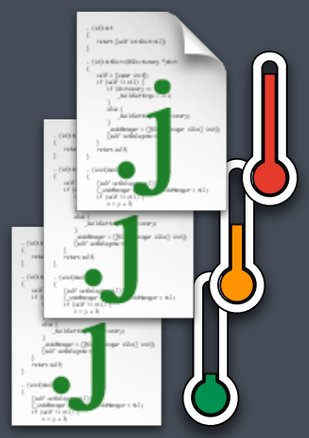
past defect per entity (quality)

Data Sources

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

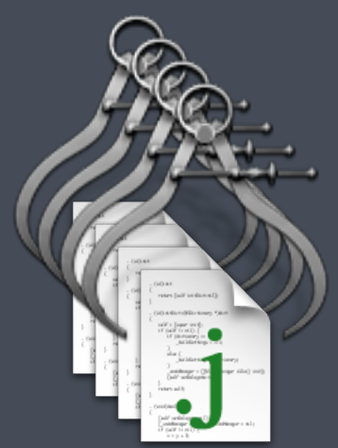
Bug Database

Version Archive



past defect per entity (quality)

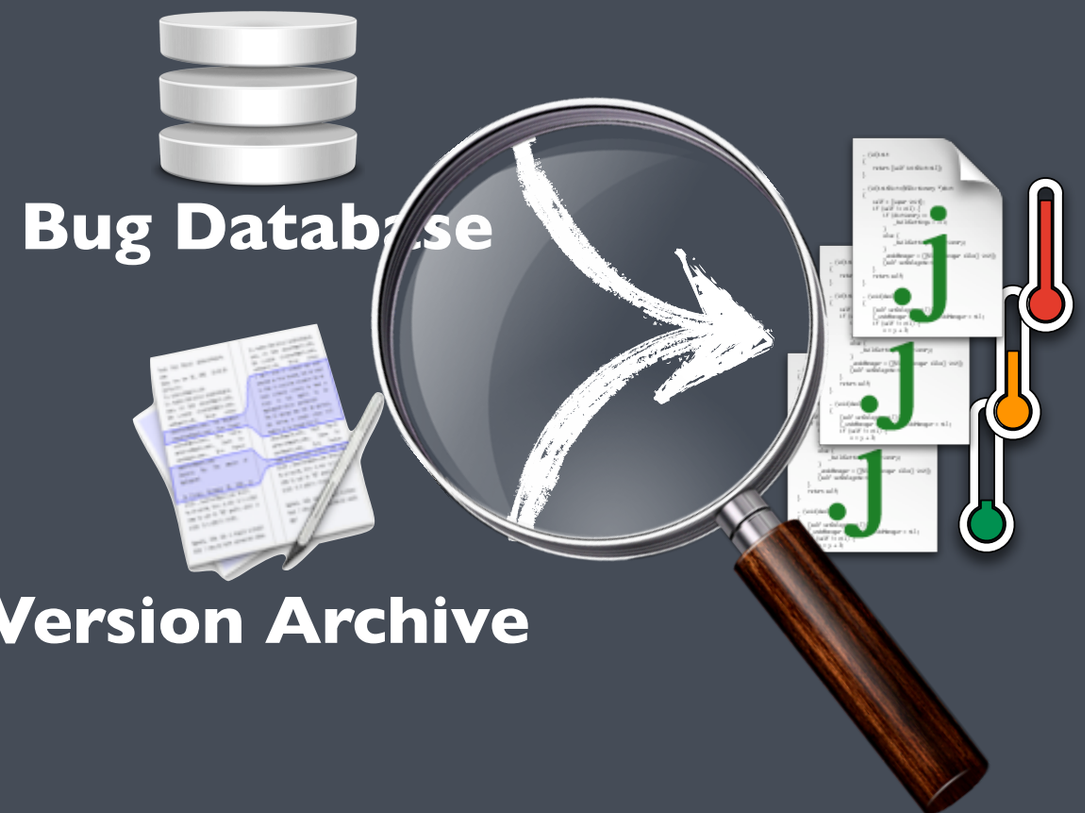
Source Code



source code properties (metrics)

Data Sources: Heuristics

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

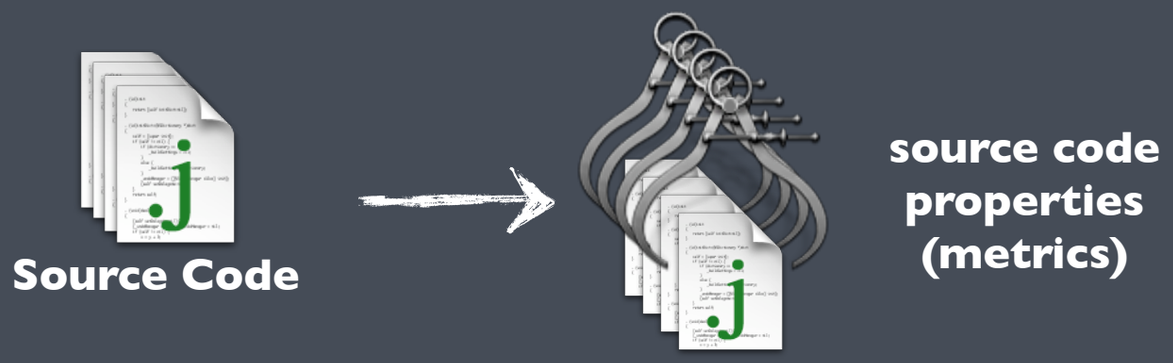


" ... commit messages that contain fix and bug id ..."

Data Sources: Metrics

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

- Complicity metrics
  - McCabe, FanIn, FanOut, Couplings
  - (see Lecture "Metrics and Estimation")
- Time metrics
  - How many changes
  - How many different authors
  - Age of code

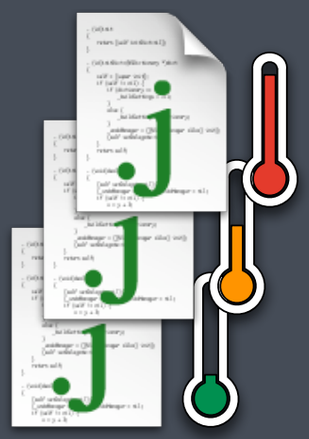


Data Sources

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

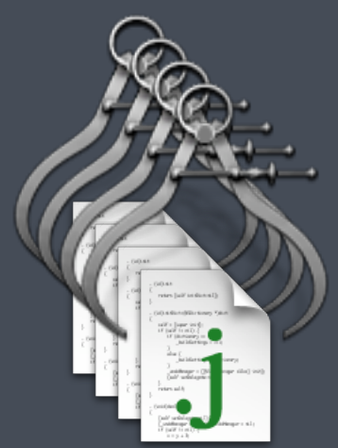
Bug Database

Version Archive



past defect per entity (quality)

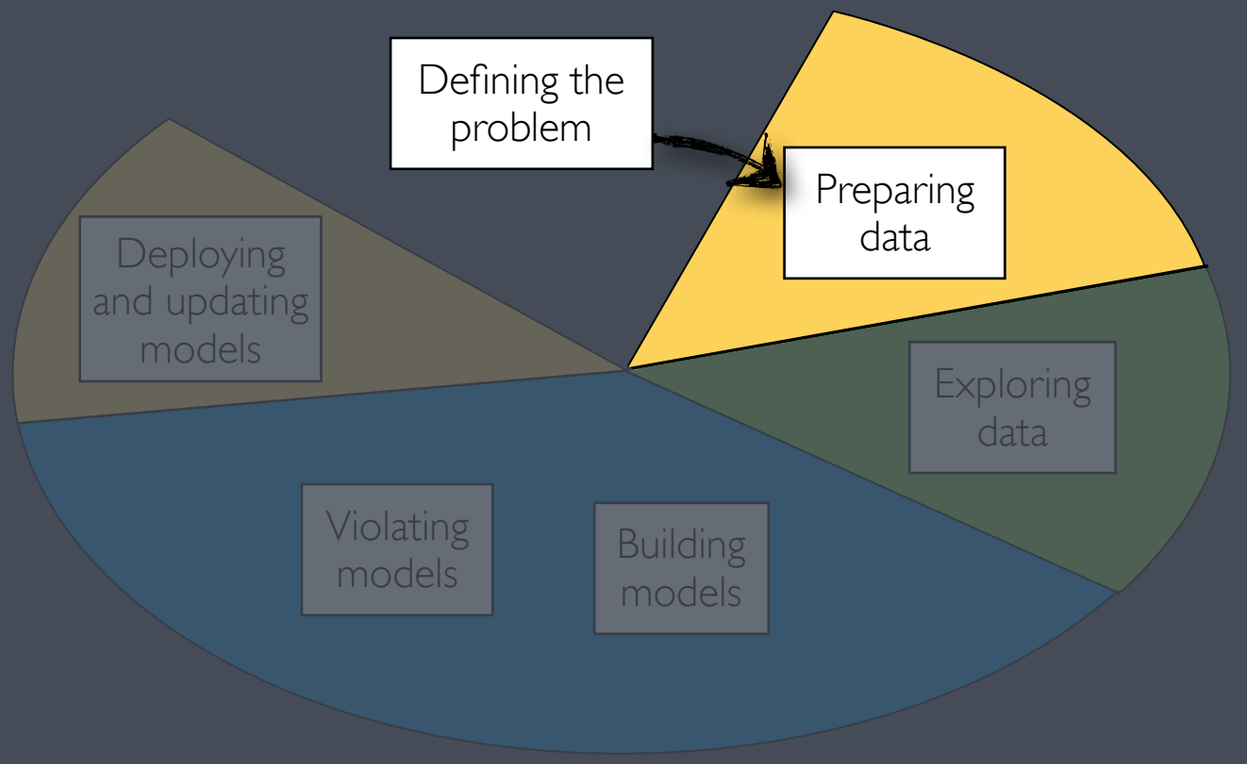
Source Code



source code properties (metrics)

Step 2: Prepare Data

- Highly distributed data:
  - Version repository, bug data base, time trackers, ...
- Data integration
  - Excel, CSV, SQL, ARFF, ...
- Data cleaning
  - missing values, noise, inter-correlations



Example Mining File

Package	Name	bugs	No. Methods	CCV	LCOM	TCC	MAXCC	AVCC	NOS	HEFF	UWCS
org.jruby.compiler	ASTCompiler	0	278	605	1	605	92	2.18	1574	424794.18	279
org.jruby.compiler	ASTCompiler$OpElementAsgnArgumentsCallback	0	3	6	0	6	3	2	21	3592.03	4
org.jruby.compiler	ASTCompiler$SpecificArityArguments	6	3	7	0.5	7	4	2.33	22	2660.26	5
org.jruby.compiler	ASTCompiler$VariableArityArguments	0	3	3	0.5	3	1	1	7	31.61	4
org.jruby.compiler	ASTCompiler19	1	20	40	0	40	10	2	117	26264.39	20
org.jruby.compiler	ASTInspector	0	18	51	1.05	51	25	2.83	433	562739.87	45
org.jruby.evaluator	ASTInterpreter	4	22	55	0	55	6	2.5	169	38462.7	22
org.jruby.runtime	AbstractCompiledBlockCallback	0	0	0	0	0	0	0	1	0	0
org.jruby.ext.ffi	AbstractInvoker	5	7	11	0.92	11	4	1.57	42	6346.91	9
org.jruby.ext.ffi	AbstractMemory	0	68	105	0.97	105	5	1.54	310	87618.03	71
org.jruby.ast.executable	AbstractScript	1	166	187	24	187	3	1.13	413	28142.16	190
org.jruby.compiler.impl	AbstractVariableCompiler	8	20	61	0.71	61	12	3.05	279	129005.27	27
org.jruby.util	Adler32Ext	0	9	10	0.67	10	2	1.11	27	484	12
org.jruby.internal.runtime.methods	AliasMethod	2	40	45	0.96	45	6	1.12	102	1383.47	53
org.jruby.ast	AliasNode	2	7	7	0.67	7	1	1	24	90.68	9
org.jruby.ext.ffi	AllocatedDirectMemoryIO	11	2	2	0	2	1	1	3	0	2
org.jruby.ext.ffi.jffi	AllocatedNativeMemoryIO	3	5	8	0.75	8	2	1.6	20	1138.3	8
org.jruby.ext.ffi.jna	AllocatedNativeMemoryIO	0	4	6	0.67	6	2	1.5	19	884.91	5
org.jruby.ast	AndNode	2	7	8	0.5	8	2	1.14	27	871.77	9
org.jruby.anno	AnnotationBinder	6	3	3	1	3	1	1	27	32.73	5
org.jruby.anno	AnnotationBinder$AnnotationBindingProcessor	2	5	14	0.88	14	6	2.8	41	2180.79	7
org.jruby.anno	AnnotationBinder$AnnotationBindingProcessor$RubyClassVisitor	0	8	66	0.86	66	22	8.25	246	230539.24	10
org.jruby.ast	ArgAuxillaryNode	8	6	6	0.8	6	1	1	17	85.47	8
org.jruby.ast	ArgsCatNode	2	7	7	0.5	7	1	1	26	806.1	9
org.jruby.ast	ArgsNoArgNode	10	2	2	0	2	1	1	8	0	2
org.jruby.ast	ArgsNode	4	34	64	0.9	64	7	1.88	146	33903.67	48
org.jruby.ast	ArgsPreOneArgNode	11	6	6	0	6	1	1	17	115.79	6
org.jruby.ast	ArgsPreTwoArgNode	2	7	8	0	8	2	1.14	20	231.46	7
org.jruby.ast	ArgsPushNode	2	7	7	0.67	7	1	1	26	642.14	9
org.jruby.ast.util	ArgsUtil	0	5	13	0	13	3	2.6	35	4082.78	5
org.jruby.ast	ArgumentNode	1	6	6	0.6	6	1	1	17	53.48	7
org.jruby.compiler	ArgumentsCallback	1	1	1	0	1	1	1	2	0	1
org.jruby.runtime	Arity	12	26	55	1.01	55	7	2.12	162	34258.83	37

Example Mining File

Package	Name	bugs	No. Methods	CCV	LCOM	TCC	MAXCC	AVCC	NOS	HEFF	UWCS
org.jruby.compiler	ASTCompiler	0	278	605	1	605	92	2.18	1574	424794.18	279
org.jruby.compiler	ASTCompiler$OpElementAsgnArgumentsCallback	0	3	6	0	6	3	2	21	3592.03	4
org.jruby.compiler	ASTCompiler$SpecificArityArguments	6	3	7	0.5	7	4	2.33	22	2660.26	5
org.jruby.compiler	ASTCompiler$VariableArityArguments	0	3	3	0.5	3	1	1	7	31.61	4
org.jruby.compiler	ASTCompiler19	1	20	40	0	40	10	2	117	26264.39	20
org.jruby.compiler	ASTInspector	0	18	51	1.05	51	25	2.83	433	562739.87	45
org.jruby.evaluator	ASTInterpreter	4	22	55	0	55	6	2.5	169	38462.7	22
org.jruby.runtime	AbstractCompiledBlockCallback	0	0	0	0	0	0	0	1	0	0
org.jruby.ext.ffi	AbstractInvoker	5	7	11	0.92	11	4	1.57	42	6346.91	9
org.jruby.ext.ffi	AbstractMemory	0	68	105	0.97	105	5	1.54	310	87618.03	71
org.jruby.ast.executable	AbstractScript	1	166	187	24	187	3	1.13	413	28142.16	190
org.jruby.compiler.impl	AbstractVariableCompiler	8	20	61	0.71	61	12	3.05	279	129005.27	27
org.jruby.util	Adler32Ext	0	9	10	0.67	10	2	1.11	27	484	12
org.jruby.internal.runtime.methods	AliasMethod	2	40	45	0.96	45	6	1.12	102	1383.47	53
org.jruby.ast	AliasNode	2	7	7	0.67	7	1	1	24	90.68	9
org.jruby.ext.ffi	AllocatedDirectMemoryIO	11	2	2	0	2	1	1	3	0	2
org.jruby.ext.ffi.jffi	AllocatedNativeMemoryIO	3	5	8	0.75	8	2	1.6	20	1138.3	8
org.jruby.ext.ffi.jna	AllocatedNativeMemoryIO	0	4	6	0.67	6	2	1.5	19	884.91	5
org.jruby.ast	AndNode	2	7	8	0.5	8	2	1.14	27	871.77	9
org.jruby.anno	AnnotationBinder	6	3	3	1	3	1	1	27	32.73	5
org.jruby.anno	AnnotationBinder$AnnotationBindingProcessor	2	5	14	0.88	14	6	2.8	41	2180.79	7
org.jruby.anno	AnnotationBinder$AnnotationBindingProcessor$RubyClassVisitor	0	8	66	0.86	66	22	8.25	246	230539.24	10
org.jruby.ast	ArgAuxillaryNode	8	6	6	0.8	6	1	1	17	85.47	8
org.jruby.ast	ArgsCatNode	2	7	7	0.5	7	1	1	26	806.1	9
org.jruby.ast	ArgsNoArgNode	10	2	2	0	2	1	1	8	0	2
org.jruby.ast	ArgsNode	4	34	64	0.9	64	7	1.88	146	33903.67	48
org.jruby.ast	ArgsPreOneArgNode	11	6	6	0	6	1	1	17	115.79	6
org.jruby.ast	ArgsPreTwoArgNode	2	7	8	0	8	2	1.14	20	231.46	7
org.jruby.ast	ArgsPushNode	2	7	7	0.67	7	1	1	26	642.14	9
org.jruby.ast.util	ArgsUtil	0	5	13	0	13	3	2.6	35	4082.78	5
org.jruby.ast	ArgumentNode	1	6	6	0.6	6	1	1	17	53.48	7
org.jruby.compiler	ArgumentsCallback	1	1	1	0	1	1	1	2	0	1
org.jruby.runtime	Arity	12	26	55	1.01	55	7	2.12	162	34258.83	37

entities

Example Mining File

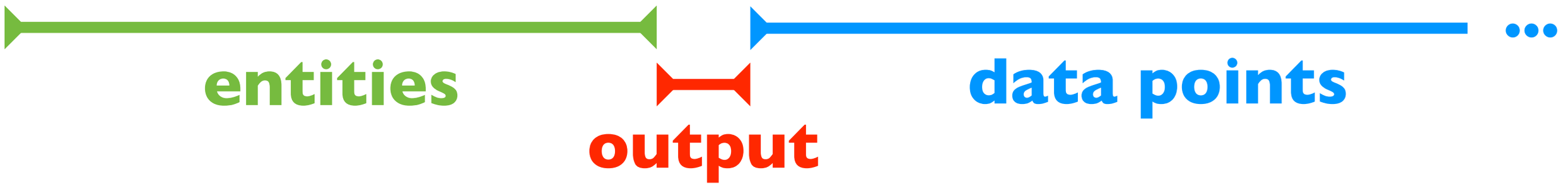
Package	Name	bugs	No. Methods	CCV	LCOM	TCC	MAXCC	AVCC	NOS	HEFF	UWCS
org.jruby.compiler	ASTCompiler	0	278	605	1	605	92	2.18	1574	424794.18	279
org.jruby.compiler	ASTCompiler$OpElementAsgnArgumentsCallback	0	3	6	0	6	3	2	21	3592.03	4
org.jruby.compiler	ASTCompiler$SpecificArityArguments	6	3	7	0.5	7	4	2.33	22	2660.26	5
org.jruby.compiler	ASTCompiler$VariableArityArguments	0	3	3	0.5	3	1	1	7	31.61	4
org.jruby.compiler	ASTCompiler19	1	20	40	0	40	10	2	117	26264.39	20
org.jruby.compiler	ASTInspector	0	18	51	1.05	51	25	2.83	433	562739.87	45
org.jruby.evaluator	ASTInterpreter	4	22	55	0	55	6	2.5	169	38462.7	22
org.jruby.runtime	AbstractCompiledBlockCallback	0	0	0	0	0	0	0	1	0	0
org.jruby.ext.ffi	AbstractInvoker	5	7	11	0.92	11	4	1.57	42	6346.91	9
org.jruby.ext.ffi	AbstractMemory	0	68	105	0.97	105	5	1.54	310	87618.03	71
org.jruby.ast.executable	AbstractScript	1	166	187	24	187	3	1.13	413	28142.16	190
org.jruby.compiler.impl	AbstractVariableCompiler	8	20	61	0.71	61	12	3.05	279	129005.27	27
org.jruby.util	Adler32Ext	0	9	10	0.67	10	2	1.11	27	484	12
org.jruby.internal.runtime.methods	AliasMethod	2	40	45	0.96	45	6	1.12	102	1383.47	53
org.jruby.ast	AliasNode	2	7	7	0.67	7	1	1	24	90.68	9
org.jruby.ext.ffi	AllocatedDirectMemoryIO	11	2	2	0	2	1	1	3	0	2
org.jruby.ext.ffi.jffi	AllocatedNativeMemoryIO	3	5	8	0.75	8	2	1.6	20	1138.3	8
org.jruby.ext.ffi.jna	AllocatedNativeMemoryIO	0	4	6	0.67	6	2	1.5	19	884.91	5
org.jruby.ast	AndNode	2	7	8	0.5	8	2	1.14	27	871.77	9
org.jruby.anno	AnnotationBinder	6	3	3	1	3	1	1	27	32.73	5
org.jruby.anno	AnnotationBinder$AnnotationBindingProcessor	2	5	14	0.88	14	6	2.8	41	2180.79	7
org.jruby.anno	AnnotationBinder$AnnotationBindingProcessor$RubyClassVisitor	0	8	66	0.86	66	22	8.25	246	230539.24	10
org.jruby.ast	ArgAuxillaryNode	8	6	6	0.8	6	1	1	17	85.47	8
org.jruby.ast	ArgsCatNode	2	7	7	0.5	7	1	1	26	806.1	9
org.jruby.ast	ArgsNoArgNode	10	2	2	0	2	1	1	8	0	2
org.jruby.ast	ArgsNode	4	34	64	0.9	64	7	1.88	146	33903.67	48
org.jruby.ast	ArgsPreOneArgNode	11	6	6	0	6	1	1	17	115.79	6
org.jruby.ast	ArgsPreTwoArgNode	2	7	8	0	8	2	1.14	20	231.46	7
org.jruby.ast	ArgsPushNode	2	7	7	0.67	7	1	1	26	642.14	9
org.jruby.ast.util	ArgsUtil	0	5	13	0	13	3	2.6	35	4082.78	5
org.jruby.ast	ArgumentNode	1	6	6	0.6	6	1	1	17	53.48	7
org.jruby.compiler	ArgumentsCallback	1	1	1	0	1	1	1	2	0	1
org.jruby.runtime	Arity	12	26	55	1.01	55	7	2.12	162	34258.83	37

entities

data points ...

Example Mining File

Package	Name	bugs	No. Methods	CCV	LCOM	TCC	MAXCC	AVCC	NOS	HEFF	UWCS
org.jruby.compiler	ASTCompiler	0	278	605	1	605	92	2.18	1574	424794.18	279
org.jruby.compiler	ASTCompiler$OpElementAsgnArgumentsCallback	0	3	6	0	6	3	2	21	3592.03	4
org.jruby.compiler	ASTCompiler$SpecificArityArguments	6	3	7	0.5	7	4	2.33	22	2660.26	5
org.jruby.compiler	ASTCompiler$VariableArityArguments	0	3	3	0.5	3	1	1	7	31.61	4
org.jruby.compiler	ASTCompiler19	1	20	40	0	40	10	2	117	26264.39	20
org.jruby.compiler	ASTInspector	0	18	51	1.05	51	25	2.83	433	562739.87	45
org.jruby.evaluator	ASTInterpreter	4	22	55	0	55	6	2.5	169	38462.7	22
org.jruby.runtime	AbstractCompiledBlockCallback	0	0	0	0	0	0	0	1	0	0
org.jruby.ext.ffi	AbstractInvoker	5	7	11	0.92	11	4	1.57	42	6346.91	9
org.jruby.ext.ffi	AbstractMemory	0	68	105	0.97	105	5	1.54	310	87618.03	71
org.jruby.ast.executable	AbstractScript	1	166	187	24	187	3	1.13	413	28142.16	190
org.jruby.compiler.impl	AbstractVariableCompiler	8	20	61	0.71	61	12	3.05	279	129005.27	27
org.jruby.util	Adler32Ext	0	9	10	0.67	10	2	1.11	27	484	12
org.jruby.internal.runtime.methods	AliasMethod	2	40	45	0.96	45	6	1.12	102	1383.47	53
org.jruby.ast	AliasNode	2	7	7	0.67	7	1	1	24	90.68	9
org.jruby.ext.ffi	AllocatedDirectMemoryIO	11	2	2	0	2	1	1	3	0	2
org.jruby.ext.ffi.jffi	AllocatedNativeMemoryIO	3	5	8	0.75	8	2	1.6	20	1138.3	8
org.jruby.ext.ffi.jna	AllocatedNativeMemoryIO	0	4	6	0.67	6	2	1.5	19	884.91	5
org.jruby.ast	AndNode	2	7	8	0.5	8	2	1.14	27	871.77	9
org.jruby.anno	AnnotationBinder	6	3	3	1	3	1	1	27	32.73	5
org.jruby.anno	AnnotationBinder$AnnotationBindingProcessor	2	5	14	0.88	14	6	2.8	41	2180.79	7
org.jruby.anno	AnnotationBinder$AnnotationBindingProcessor$RubyClassVisitor	0	8	66	0.86	66	22	8.25	246	230539.24	10
org.jruby.ast	ArgAuxillaryNode	8	6	6	0.8	6	1	1	17	85.47	8
org.jruby.ast	ArgsCatNode	2	7	7	0.5	7	1	1	26	806.1	9
org.jruby.ast	ArgsNoArgNode	10	2	2	0	2	1	1	8	0	2
org.jruby.ast	ArgsNode	4	34	64	0.9	64	7	1.88	146	33903.67	48
org.jruby.ast	ArgsPreOneArgNode	11	6	6	0	6	1	1	17	115.79	6
org.jruby.ast	ArgsPreTwoArgNode	2	7	8	0	8	2	1.14	20	231.46	7
org.jruby.ast	ArgsPushNode	2	7	7	0.67	7	1	1	26	642.14	9
org.jruby.ast.util	ArgsUtil	0	5	13	0	13	3	2.6	35	4082.78	5
org.jruby.ast	ArgumentNode	1	6	6	0.6	6	1	1	17	53.48	7
org.jruby.compiler	ArgumentsCallback	1	1	1	0	1	1	1	2	0	1
org.jruby.runtime	Arity	12	26	55	1.01	55	7	2.12	162	34258.83	37



Example Mining File

Package	Name	bugs	No. Methods	CCV	LCOM	TCC	MAXCC	AVCC	NOS	HEFF	UWCS
org.jruby.compiler	ASTCompiler	0	278	605	1	605	92	2.18	1574	424794.18	279
org.jruby.compiler	ASTCompiler$OpElementAsgnArgumentsCallback	0	3	6	0	6	3	2	21	3592.03	4
org.jruby.compiler	ASTCompiler$SpecificArityArguments	6	3	7	0.5	7	4	2.33	22	2660.26	5
org.jruby.compiler	ASTCompiler$VariableArityArguments	0	3	3	0.5	3	1	1	7	31.61	4
org.jruby.compiler	ASTCompiler19	1	20	40	0	40	2	2	7	26264.39	20
org.jruby.compiler	ASTInspector	0	18	51	1.05	25	25	2.83	3	562739.87	45
org.jruby.evaluator	ASTInterpreter	4	22	55	0	55	6	2.5	9	38462.7	22
org.jruby.runtime	AbstractCompiledBlockCallback	0	0	0	0	0	0	0	0	0	0
org.jruby.ext.ffi	AbstractInvoker	5	7	11	0.92	11	4	1.57	4	6346.91	9
org.jruby.ext.ffi	AbstractMemory	0	58	105	0.97	105	5	1.54	31	87618.03	71
org.jruby.ast.executable	AbstractScript	1	166	187	0.96	187	3	1.13	41	28142.16	190
org.jruby.compiler.impl	AbstractVariableCompiler	8	20	61	0.92	61	12	2.75	279	129005.27	27
org.jruby.util	Adler32Ext	0	10	10	0.67	10	2	1.5	22	484	12
org.jruby.internal.runtime.methods	AliasMethod	0	40	45	0.86	45	5	2.12	1	1383.47	53
org.jruby.ast	AliasNode	0	7	7	1	7	1	1	24	90.68	9
org.jruby.ext.ffi	AllocatedDirectMemoryIO	11	2	2	0	2	1	1	3	0	2
org.jruby.ext.ffi.jffi	AllocatedNativeMemoryIO	3	5	8	0.67	8	2	1.6	20	1138.3	8
org.jruby.ext.ffi.jna	AllocatedNativeMemoryIO	0	4	5	0.67	5	2	1.5	19	884.91	5
org.jruby.ast	AndNode	0	8	8	0	8	2	1.14	27	871.77	9
org.jruby.anno	AnnotationBinder	3	3	3	1	3	1	1	27	32.73	5
org.jruby.anno	AnnotationBinder$AnnotationBindingProcessor	2	14	14	0.88	14	6	2.8	41	2180.79	7
org.jruby.anno	AnnotationBinder$AnnotationBinding	9	8	66	0.86	66	22	8.25	246	230539.24	10
org.jruby.ast	ArgAuxillaryNode	8	6	6	0.8	6	1	1	17	85.47	8
org.jruby.ast	ArgsCatNode	2	7	7	0.5	7	1	1	26	806.1	9
org.jruby.ast	ArgsNoArgNode	10	2	2	0	2	1	1	8	0	2
org.jruby.ast	ArgsNode	4	34	64	0.9	64	7	1.88	146	33903.67	48
org.jruby.ast	ArgsPreOneArgNode	11	6	6	0	6	1	1	17	115.79	6
org.jruby.ast	ArgsRestArgNode	2	7	8	0	8	2	1.14	20	231.46	7
org.jruby.ast	ArgsPushNode	2	7	7	0.67	7	1	1	26	642.14	9
org.jruby.ast.util	ArgsUtil	0	5	13	0	13	3	2.6	35	4082.78	5
org.jruby.ast	ArgumentNode	1	6	6	0.6	6	1	1	17	53.48	7
org.jruby.compiler	ArgumentsCallback	1	1	1	0	1	1	1	2	0	1
org.jruby.runtime	Arity	12	26	55	1.01	55	7	2.12	162	34258.83	37

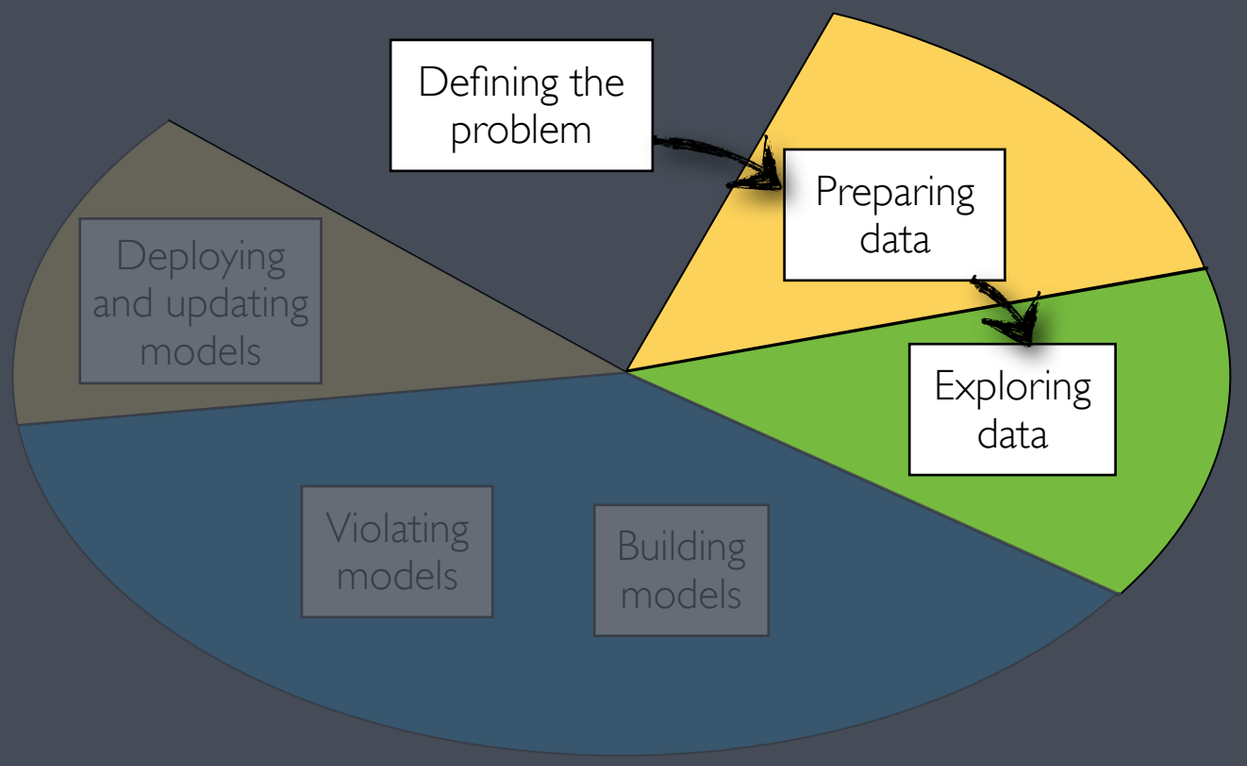
Careful! Large files!
e.g.: 5 million lines, 300 column

entities | **output** | **data points** ...

Step 3: Explore Data

You cannot validate the output if you don't know the input

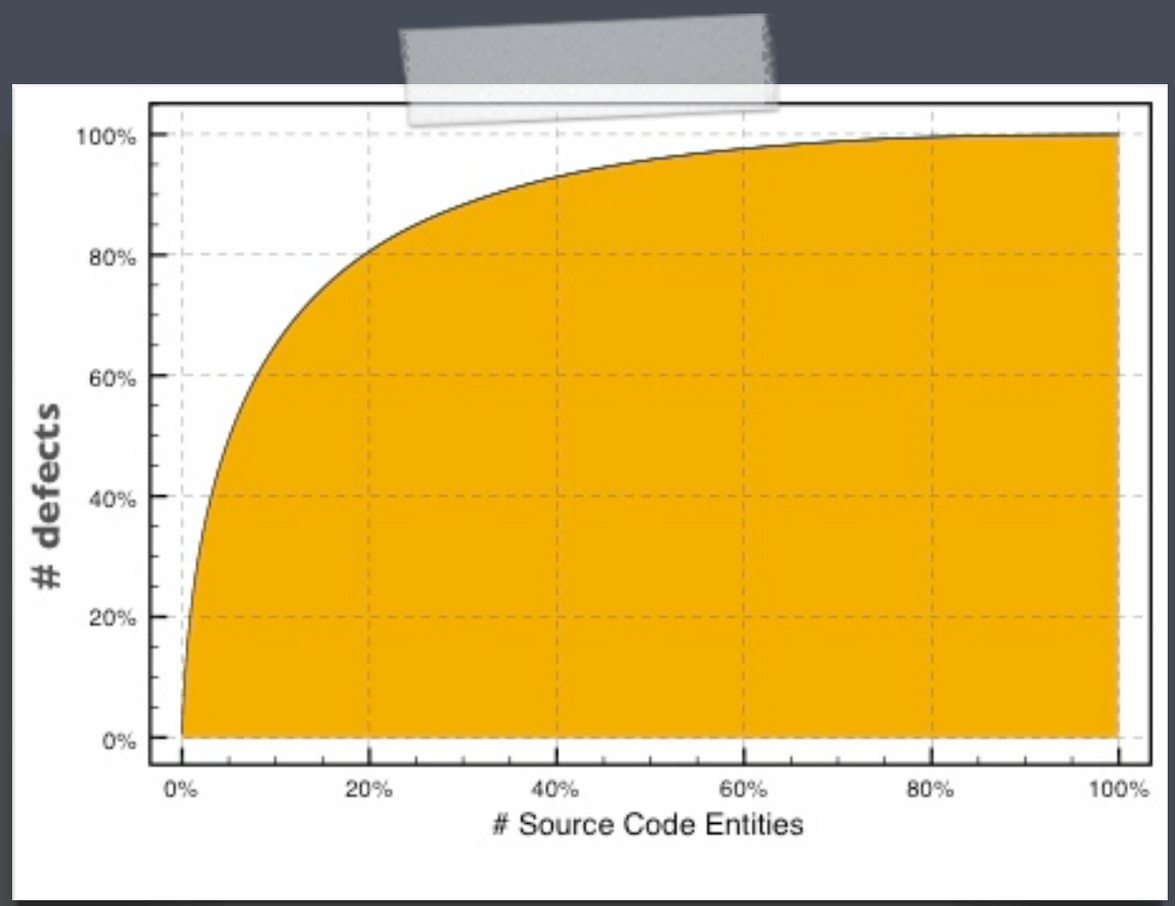
- Descriptive data summary
  - max, min, mean, pareto, distribution
- Data Selection
  - Relevance of data
- Data reduction
  - aggregation, subset selection



Descriptive Data Summary

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

- How good can a prediction possibly be?
- Does it make sense to predict the top 20%



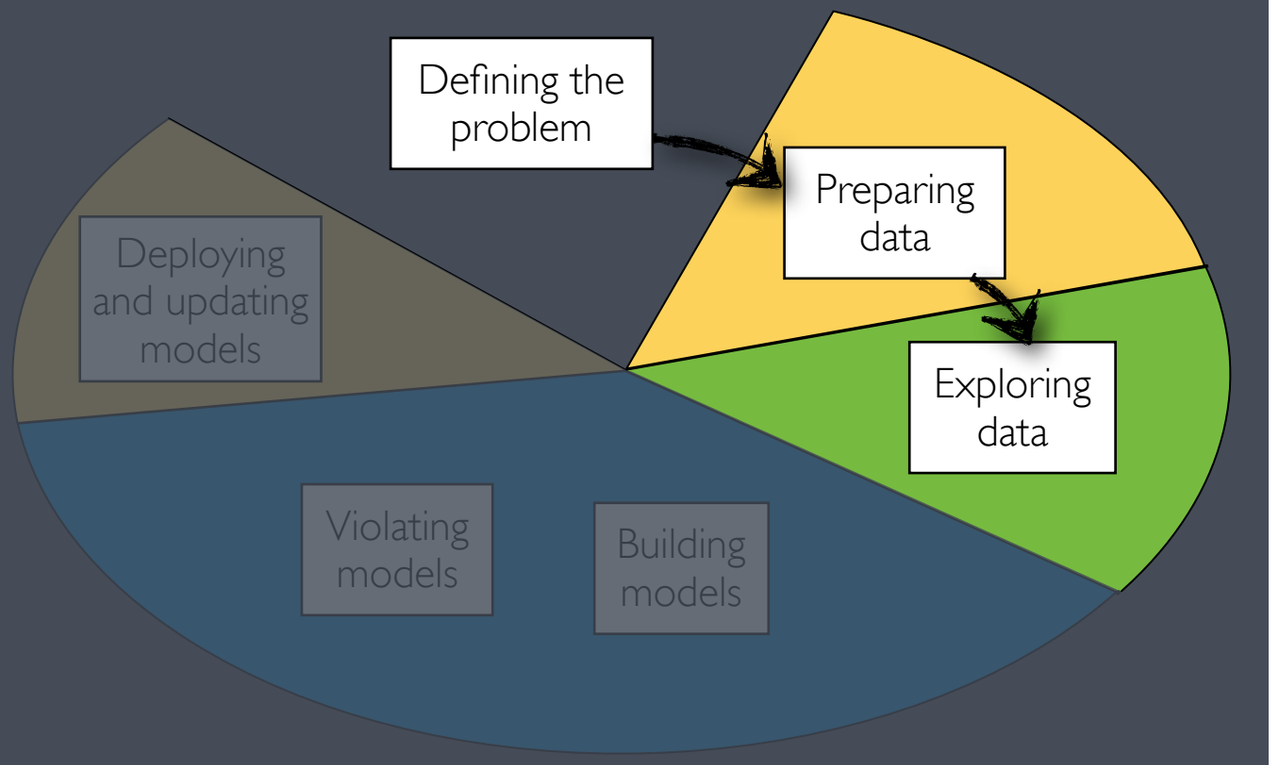
20% of entities contain 80% of defects

Step 3: Explore Data

Data sufficiency

- Maybe the data will not help to solve the problem

- Redefine problem
- Search for alternatives
- Access different data

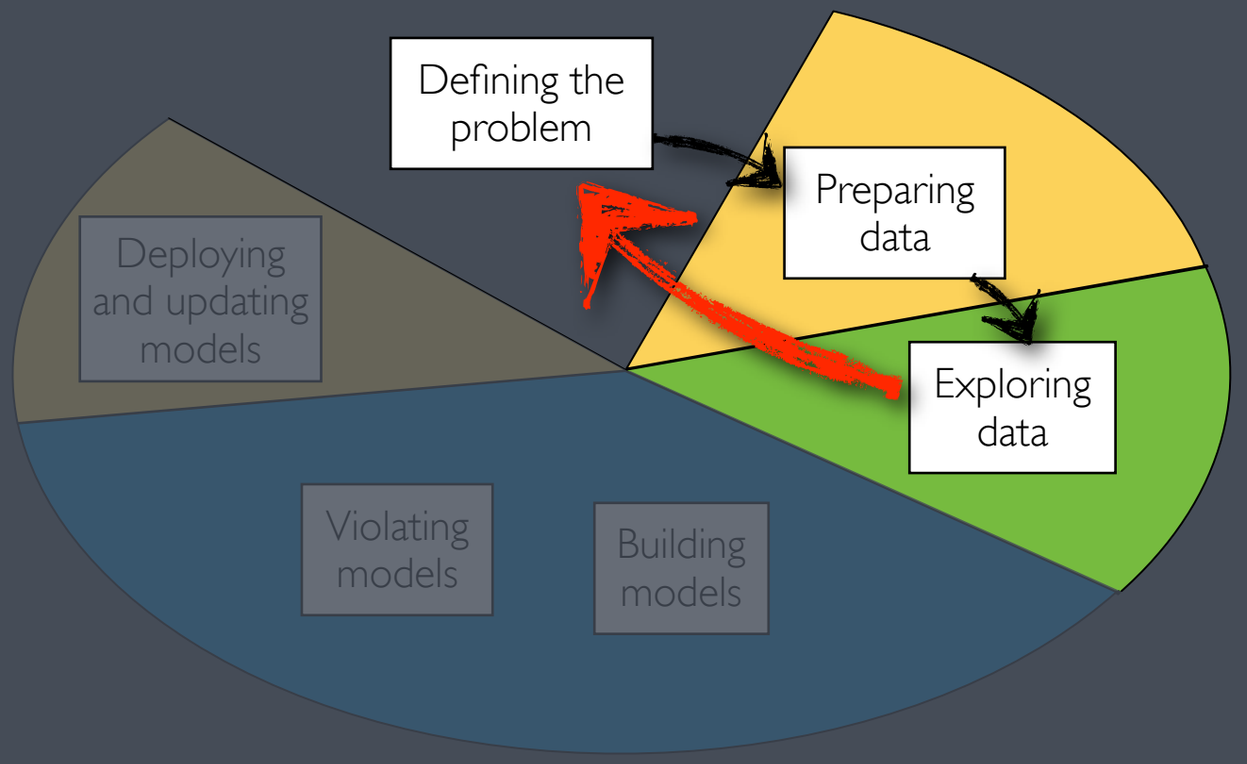


Step 3: Explore Data

Data sufficiency

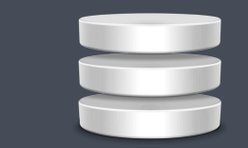
- Maybe the data will not help to solve the problem

- Redefine problem
- Search for alternatives
- Access different data



Step 3: Explore Data

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

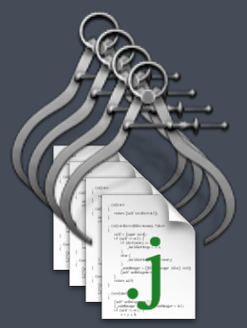


Bug Database

Version Archive

past defect
per entity (quality)

Source Code

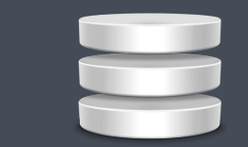


source code
properties
(metrics)

Does complexity
correlate with defects?

Step 3: Explore Data

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

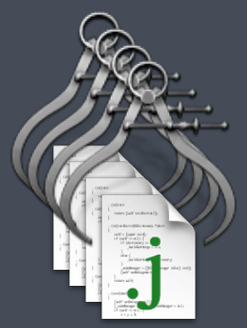


Bug Database

Version Archive

past defect
per entity (quality)

Source Code



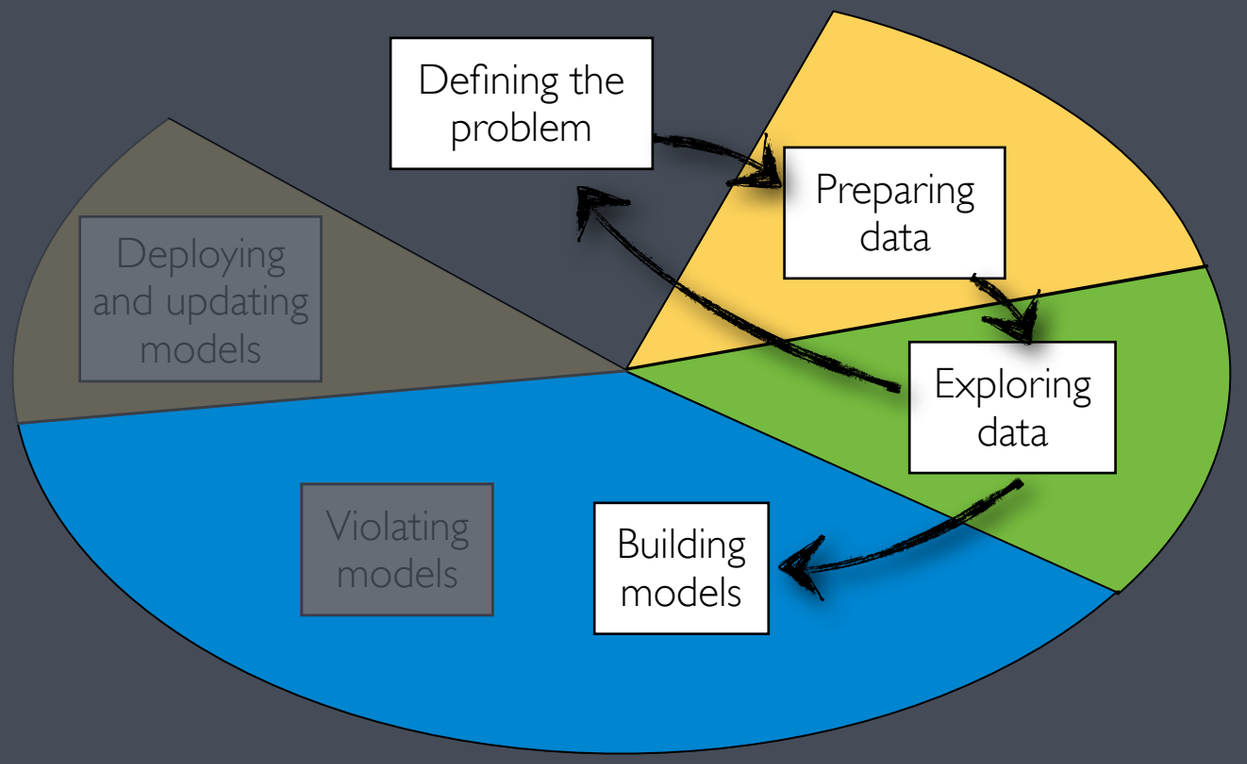
source code
properties
(metrics)

Does complexity correlate with defects?

YES!

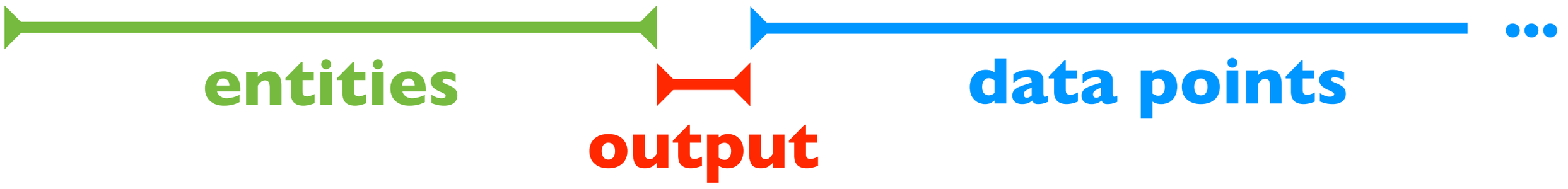
Step 4: Build Model

- Mining model only container
  - parameters and mining structure
  - output value
- Now we need some statistics / machine learners



Example Mining File

Package	Name	bugs	No. Methods	CCV	LCOM	TCC	MAXCC	AVCC	NOS	HEFF	UWCS
org.jruby.compiler	ASTCompiler	0	278	605	1	605	92	2.18	1574	424794.18	279
org.jruby.compiler	ASTCompiler$OpElementAsgnArgumentsCallback	0	3	6	0	6	3	2	21	3592.03	4
org.jruby.compiler	ASTCompiler$SpecificArityArguments	6	3	7	0.5	7	4	2.33	22	2660.26	5
org.jruby.compiler	ASTCompiler$VariableArityArguments	0	3	3	0.5	3	1	1	7	31.61	4
org.jruby.compiler	ASTCompiler19	1	20	40	0	40	10	2	117	26264.39	20
org.jruby.compiler	ASTInspector	0	18	51	1.05	51	25	2.83	433	562739.87	45
org.jruby.evaluator	ASTInterpreter	4	22	55	0	55	6	2.5	169	38462.7	22
org.jruby.runtime	AbstractCompiledBlockCallback	0	0	0	0	0	0	0	1	0	0
org.jruby.ext.ffi	AbstractInvoker	5	7	11	0.92	11	4	1.57	42	6346.91	9
org.jruby.ext.ffi	AbstractMemory	0	68	105	0.97	105	5	1.54	310	87618.03	71
org.jruby.ast.executable	AbstractScript	1	166	187	24	187	3	1.13	413	28142.16	190
org.jruby.compiler.impl	AbstractVariableCompiler	8	20	61	0.71	61	12	3.05	279	129005.27	27
org.jruby.util	Adler32Ext	0	9	10	0.67	10	2	1.11	27	484	12
org.jruby.internal.runtime.methods	AliasMethod	2	40	45	0.96	45	6	1.12	102	1383.47	53
org.jruby.ast	AliasNode	2	7	7	0.67	7	1	1	24	90.68	9
org.jruby.ext.ffi	AllocatedDirectMemoryIO	11	2	2	0	2	1	1	3	0	2
org.jruby.ext.ffi.jffi	AllocatedNativeMemoryIO	3	5	8	0.75	8	2	1.6	20	1138.3	8
org.jruby.ext.ffi.jna	AllocatedNativeMemoryIO	0	4	6	0.67	6	2	1.5	19	884.91	5
org.jruby.ast	AndNode	2	7	8	0.5	8	2	1.14	27	871.77	9
org.jruby.anno	AnnotationBinder	6	3	3	1	3	1	1	27	32.73	5
org.jruby.anno	AnnotationBinder$AnnotationBindingProcessor	2	5	14	0.88	14	6	2.8	41	2180.79	7
org.jruby.anno	AnnotationBinder$AnnotationBindingProcessor$RubyClassVisitor	0	8	66	0.86	66	22	8.25	246	230539.24	10
org.jruby.ast	ArgAuxillaryNode	8	6	6	0.8	6	1	1	17	85.47	8
org.jruby.ast	ArgsCatNode	2	7	7	0.5	7	1	1	26	806.1	9
org.jruby.ast	ArgsNoArgNode	10	2	2	0	2	1	1	8	0	2
org.jruby.ast	ArgsNode	4	34	64	0.9	64	7	1.88	146	33903.67	48
org.jruby.ast	ArgsPreOneArgNode	11	6	6	0	6	1	1	17	115.79	6
org.jruby.ast	ArgsPreTwoArgNode	2	7	8	0	8	2	1.14	20	231.46	7
org.jruby.ast	ArgsPushNode	2	7	7	0.67	7	1	1	26	642.14	9
org.jruby.ast.util	ArgsUtil	0	5	13	0	13	3	2.6	35	4082.78	5
org.jruby.ast	ArgumentNode	1	6	6	0.6	6	1	1	17	53.48	7
org.jruby.compiler	ArgumentsCallback	1	1	1	0	1	1	1	2	0	1
org.jruby.runtime	Arity	12	26	55	1.01	55	7	2.12	162	34258.83	37



Building the Model

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

- **Regression**
  - Predicting concrete, continuous values
  - Difficult and very imprecise
  - But desirable
- **Classification**
  - Predicting class labels (e.g. more that X defects or not)
  - Easier and more precise
  - Vague information (how many defects in code?)

Building the Model

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

Building the Model

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

Rule-Based Classification

Support Vector Machine

Lazy Learners

Linear Regression

Bayesian Network

Decision Tree

Logistic Regression

Training and Testing

- **Training set**
  - The data set to train the model
  - Which columns correlate with output values?
  - Which columns correlate with each other?
- **Testing set**
  - A data set independent of the training data set
  - used to fine-tune the estimates of the model parameters

Training and Testing

Random split

+ **Only one version needed**
+ **No overlaps between training and testing entities**

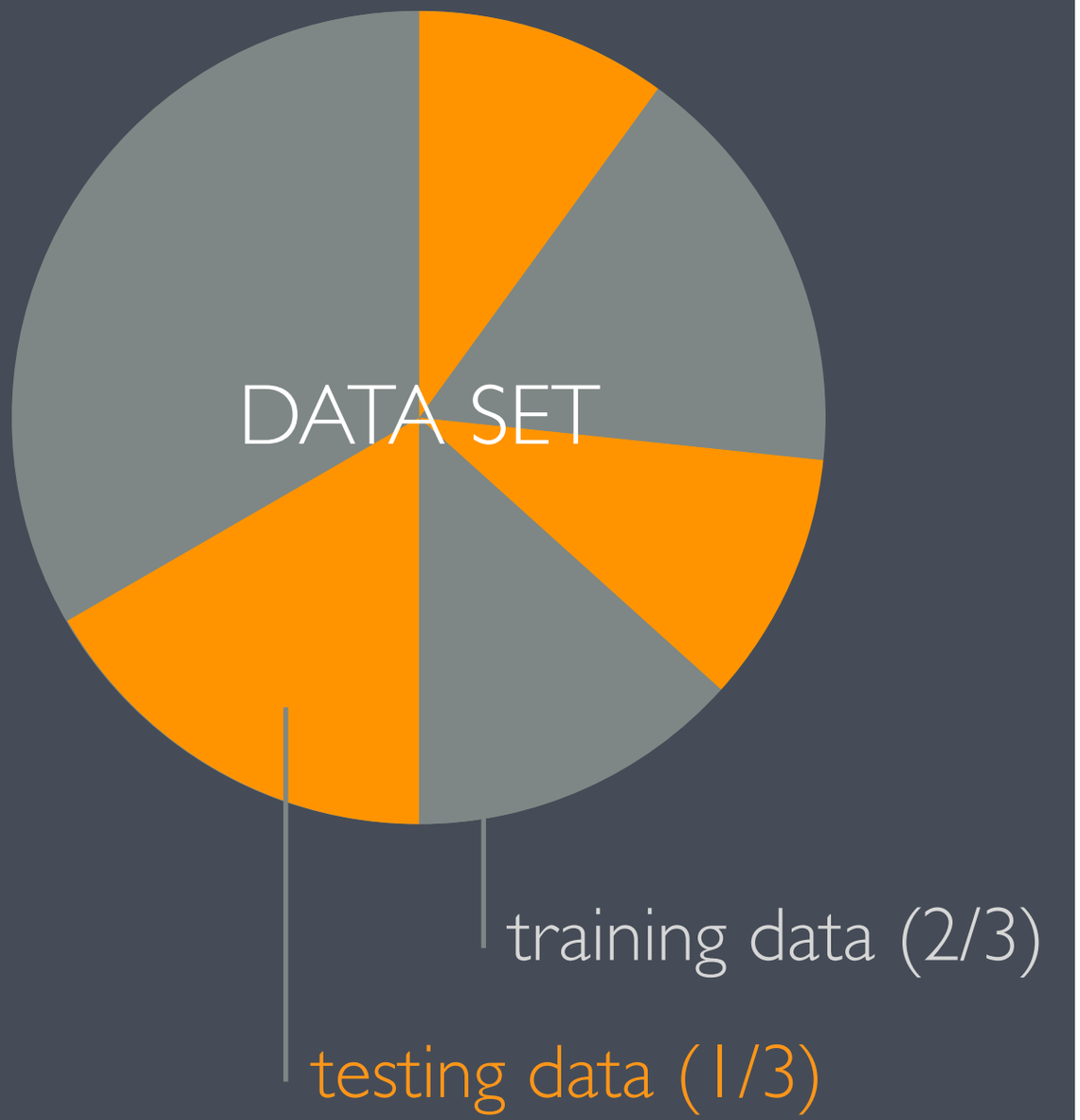
- **Does not reflect real life**
- **Which random set is the best one?** **(because they are all different)**

Training and Testing

Random split

+ **Only one version needed**
+ **No overlaps between training and testing entities**

– **Does not reflect real life**
– **Which random set is the best one?** **(because they are all different)**

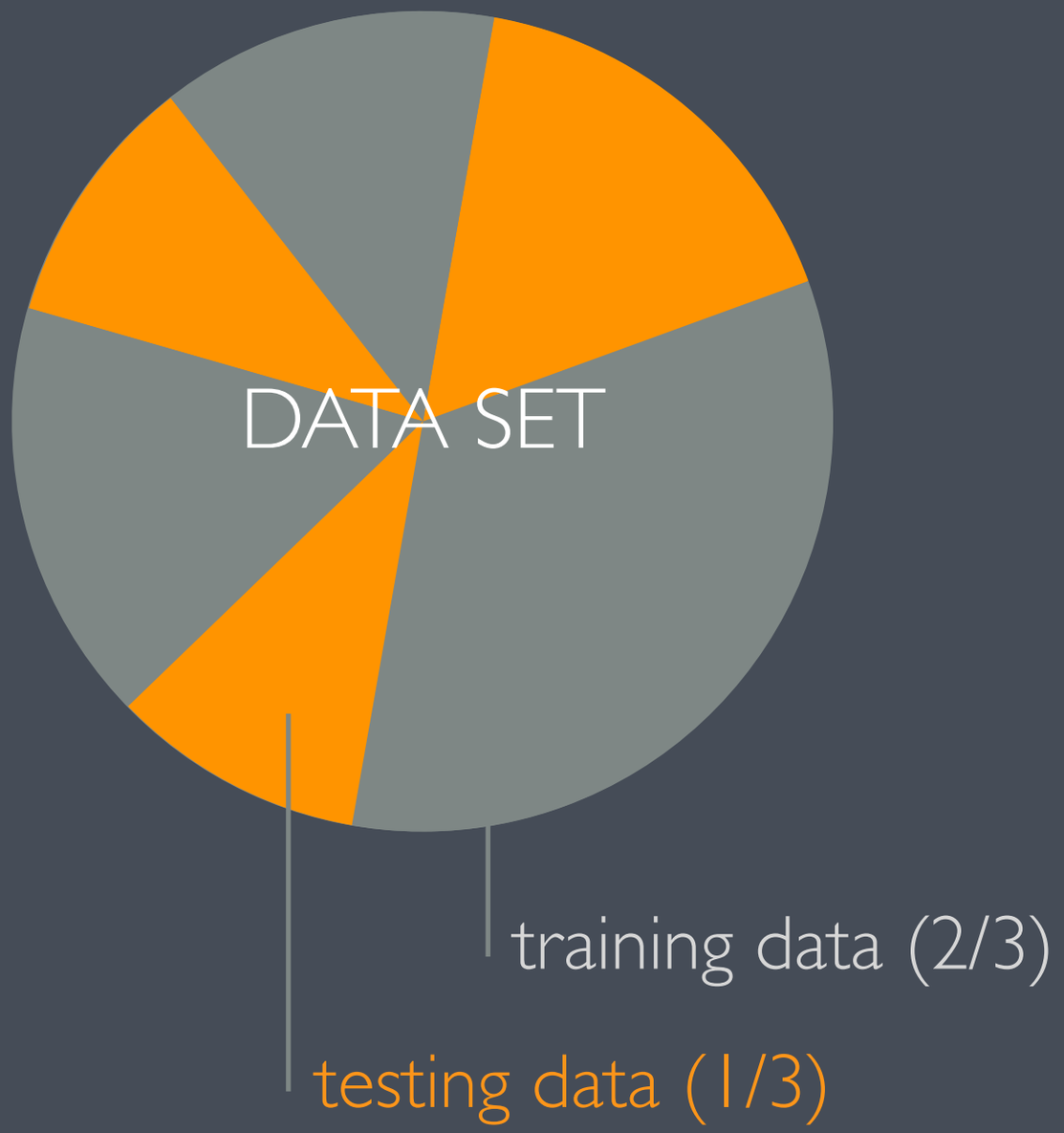


Training and Testing

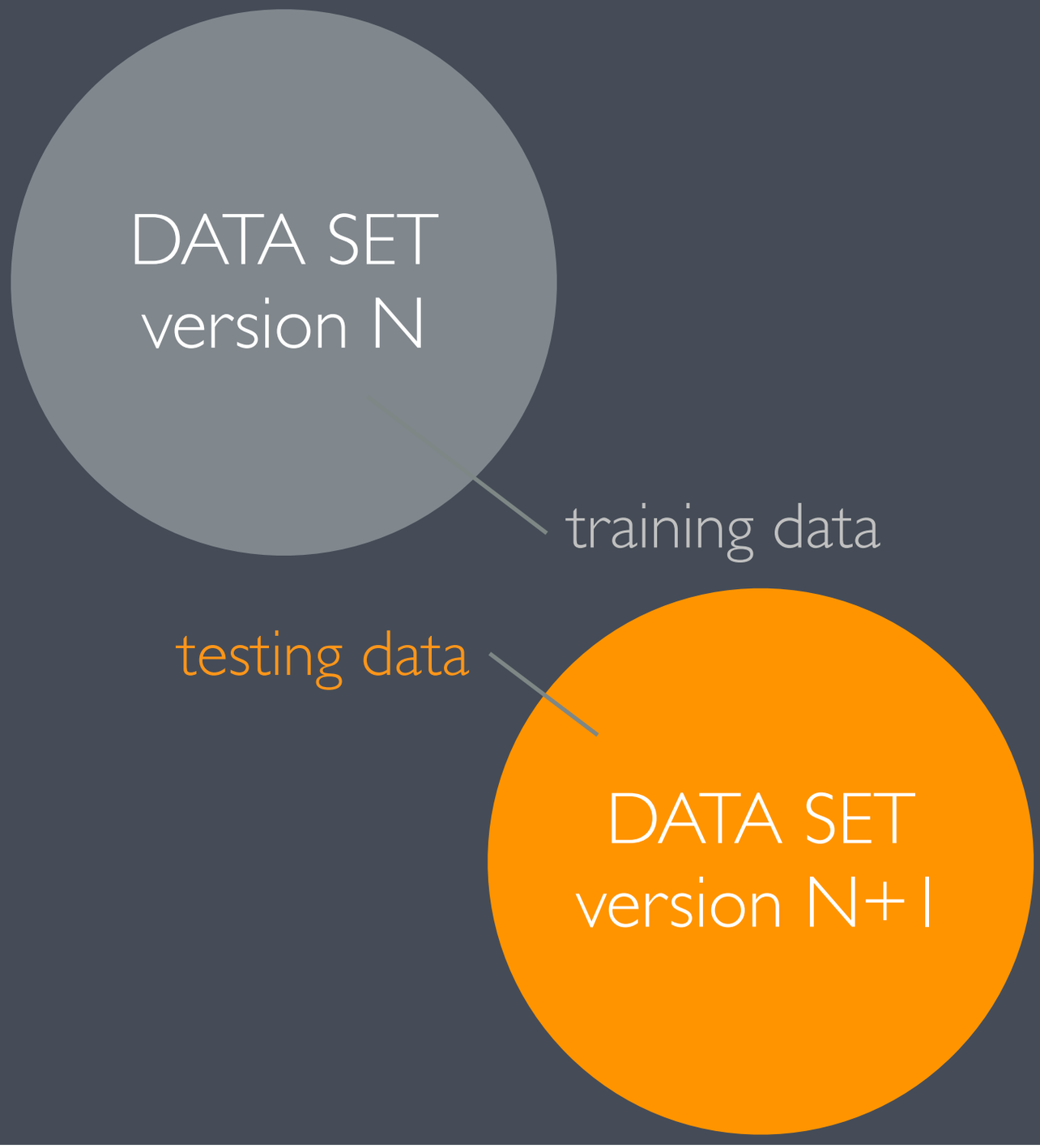
Random split

+ **Only one version needed**
+ **No overlaps between training and testing entities**

– **Does not reflect real life**
– **Which random set is the best one? (because they are all different)**



Training and Testing



Forward estimation

+ Reflects real life

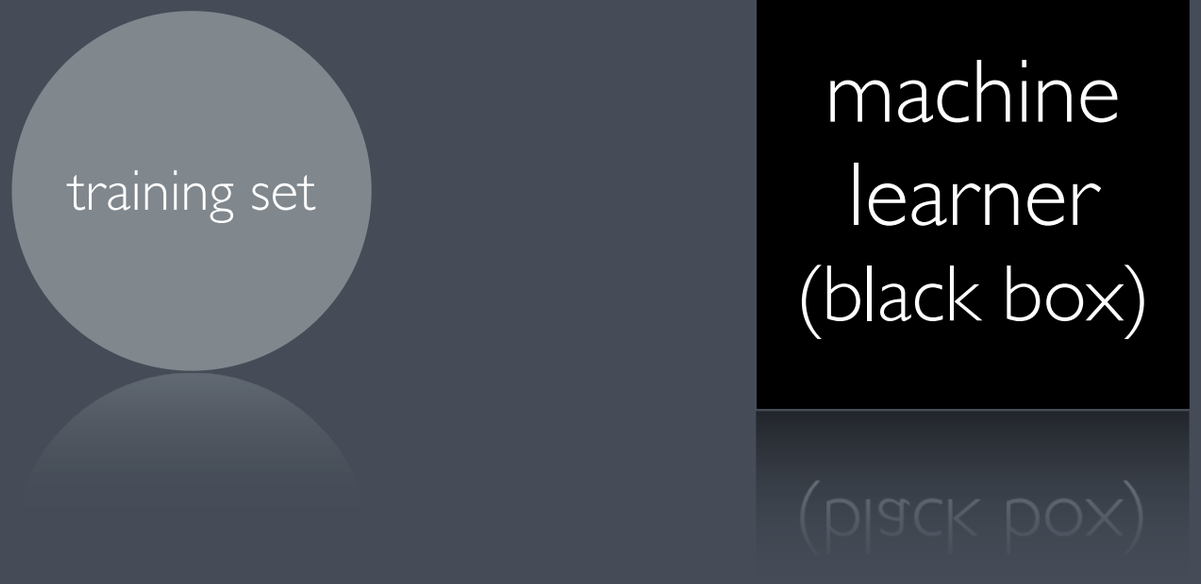
+ Reproducable result

– Two versions needed

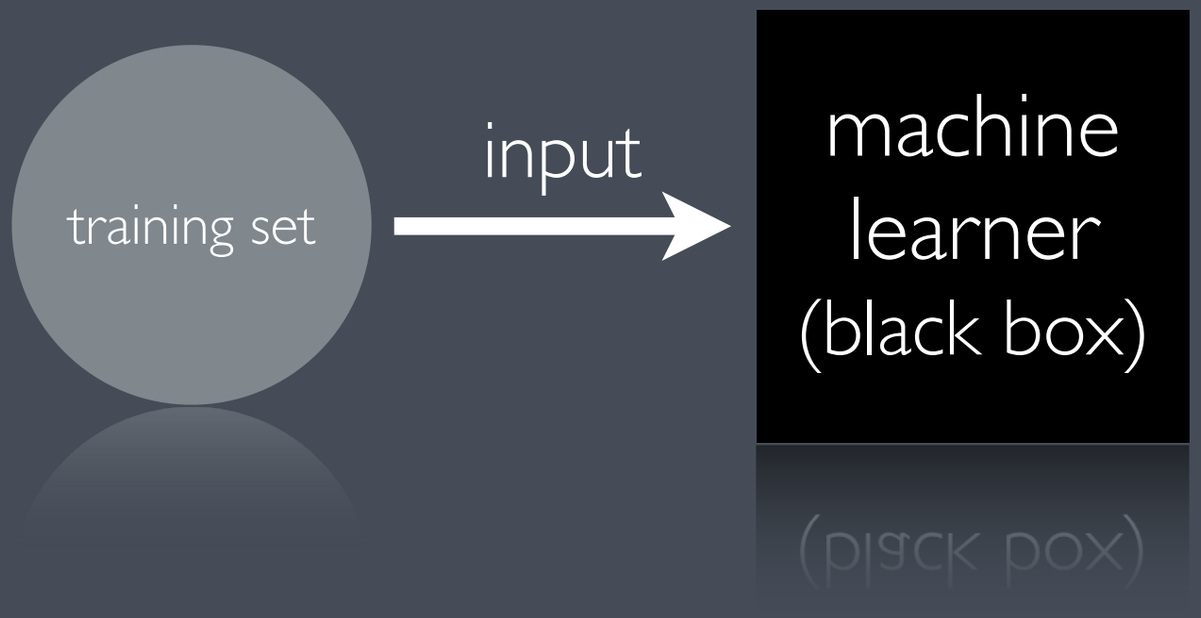
Step 4: Build Model

Step 4: Build Model

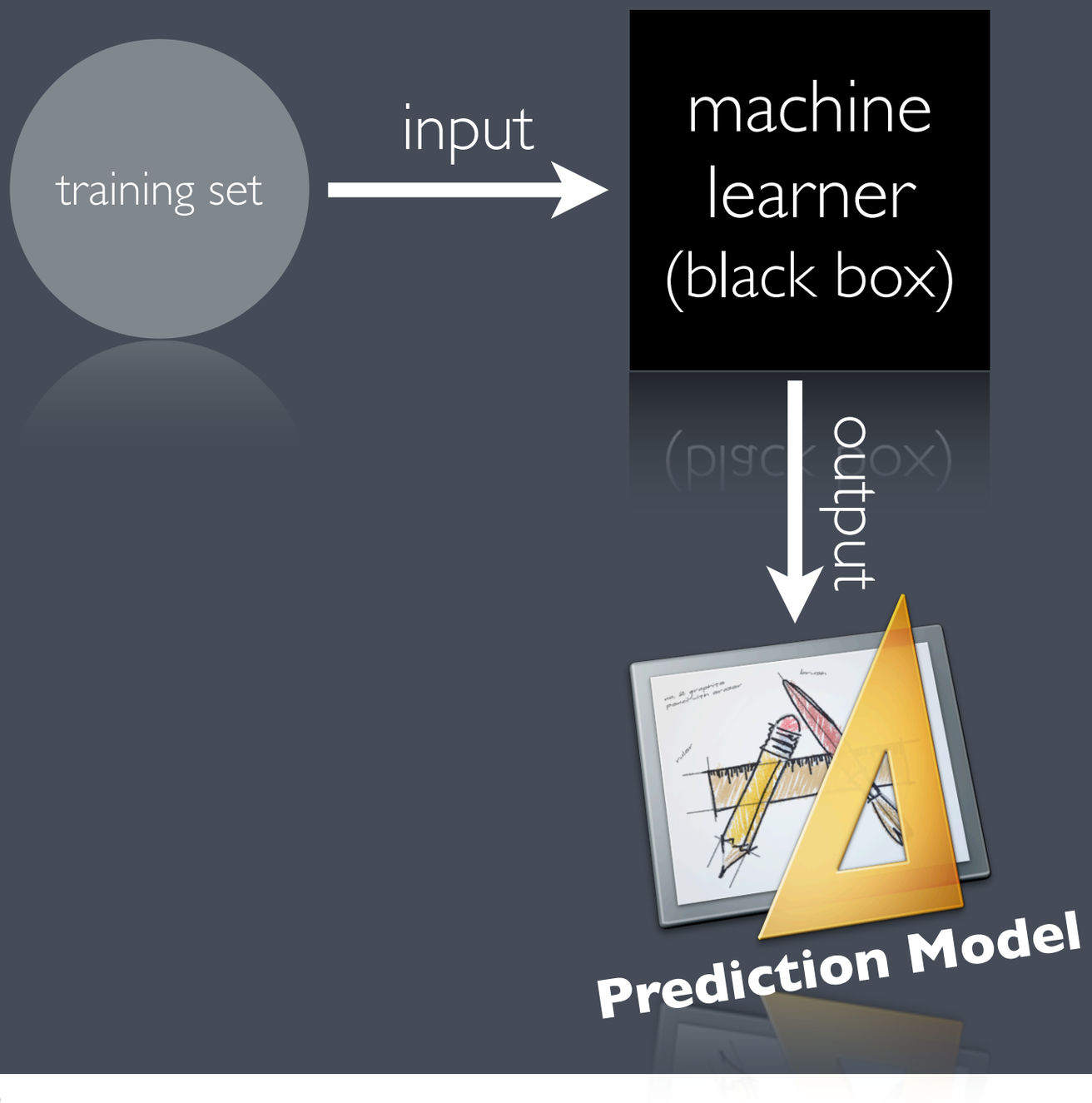
Step 4: Build Model



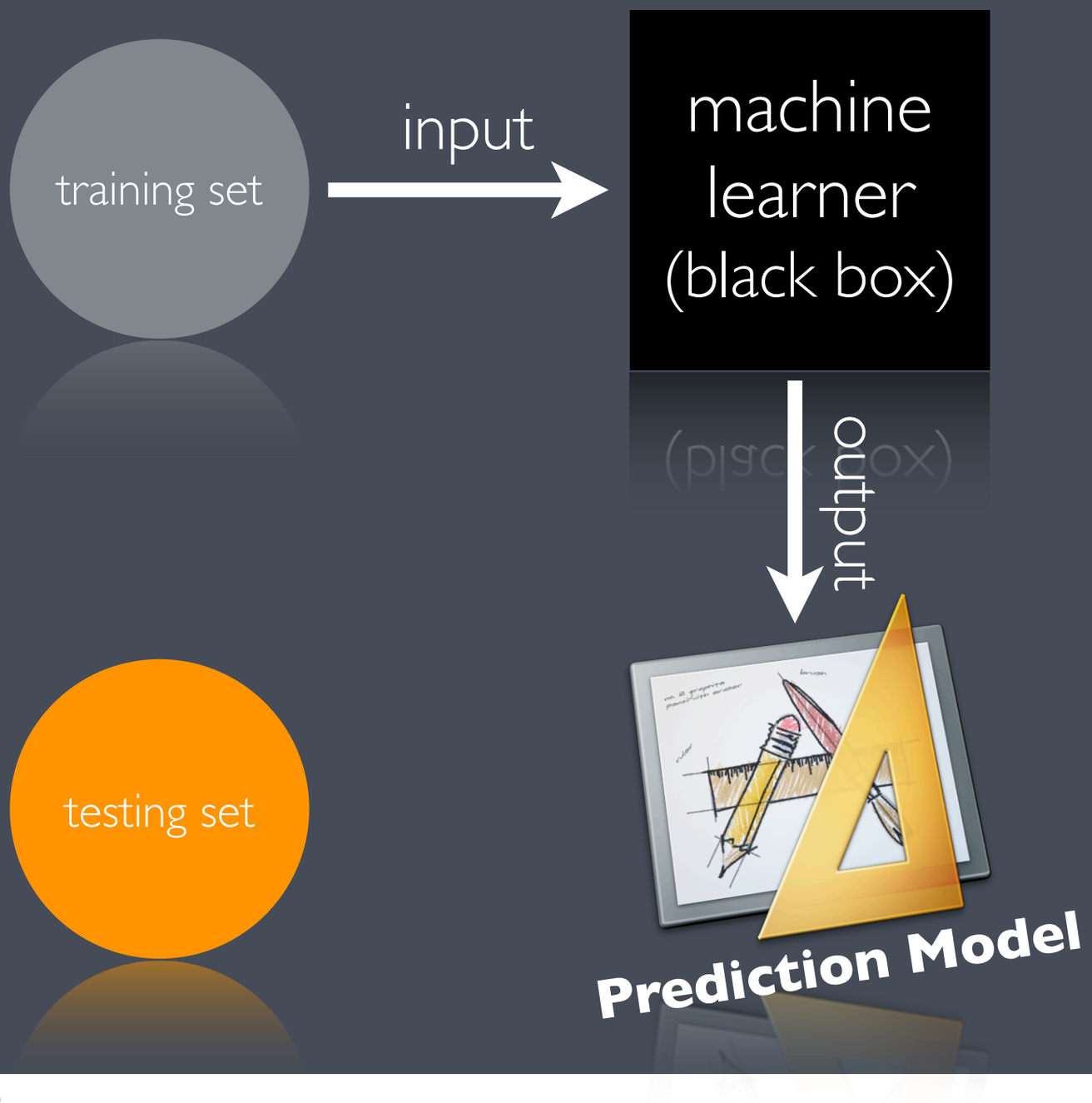
Step 4: Build Model



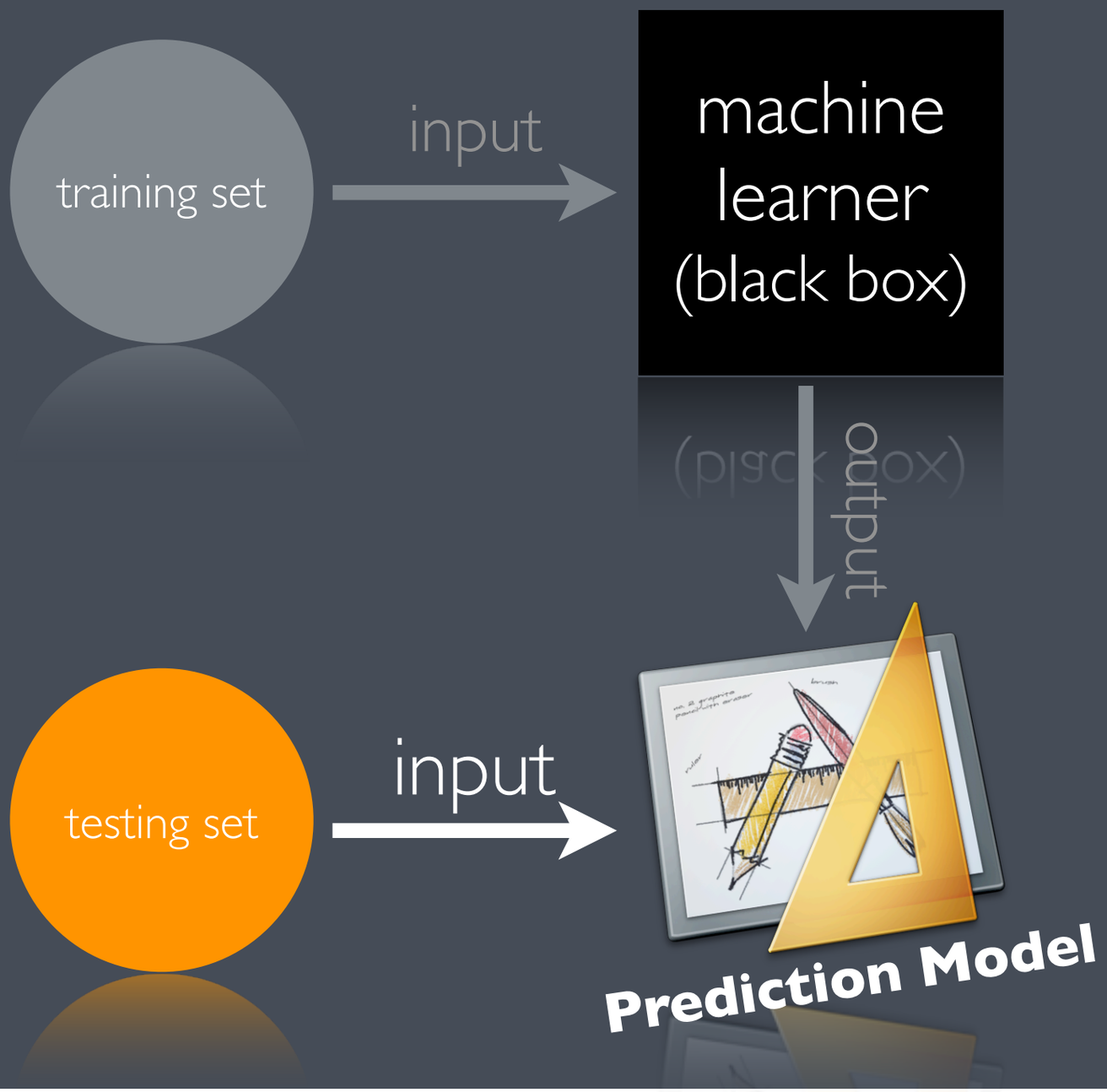
Step 4: Build Model



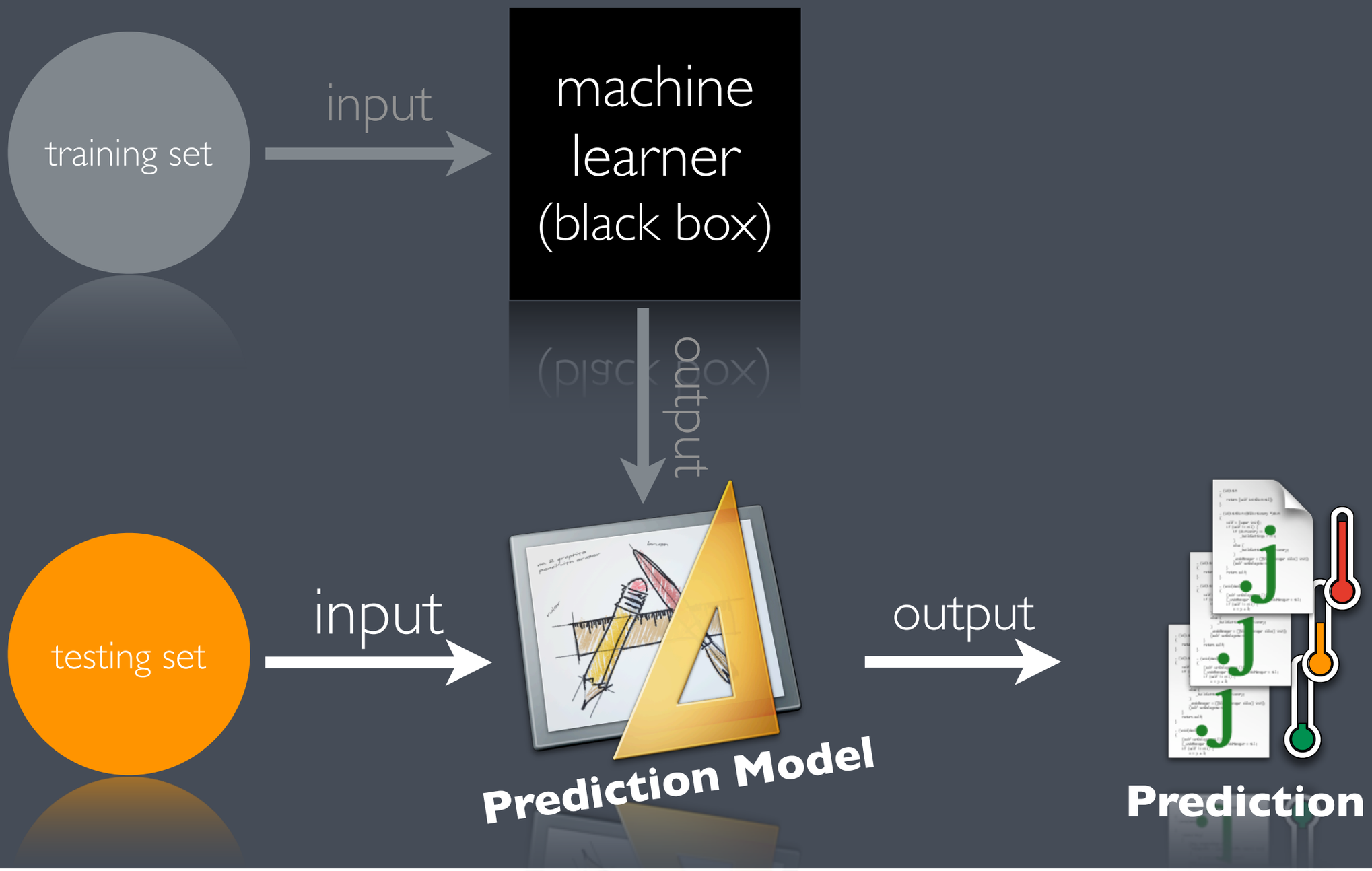
Step 4: Build Model



Step 4: Build Model



Step 4: Build Model

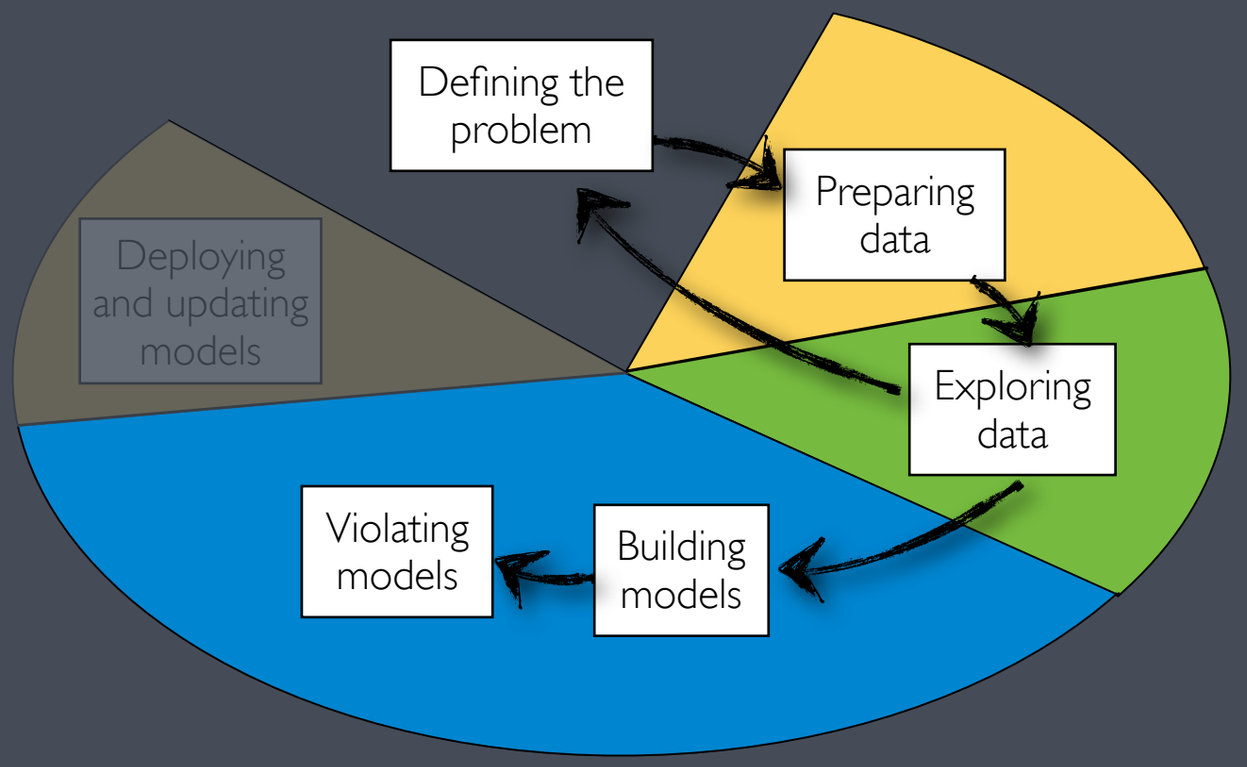


Step 5: Validating Model

- Test data has same stucture but different content

- Goal is to use model to correctly estimate output values

- Compare estimation with real values (fine tuning)



Evaluation

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

Evaluation

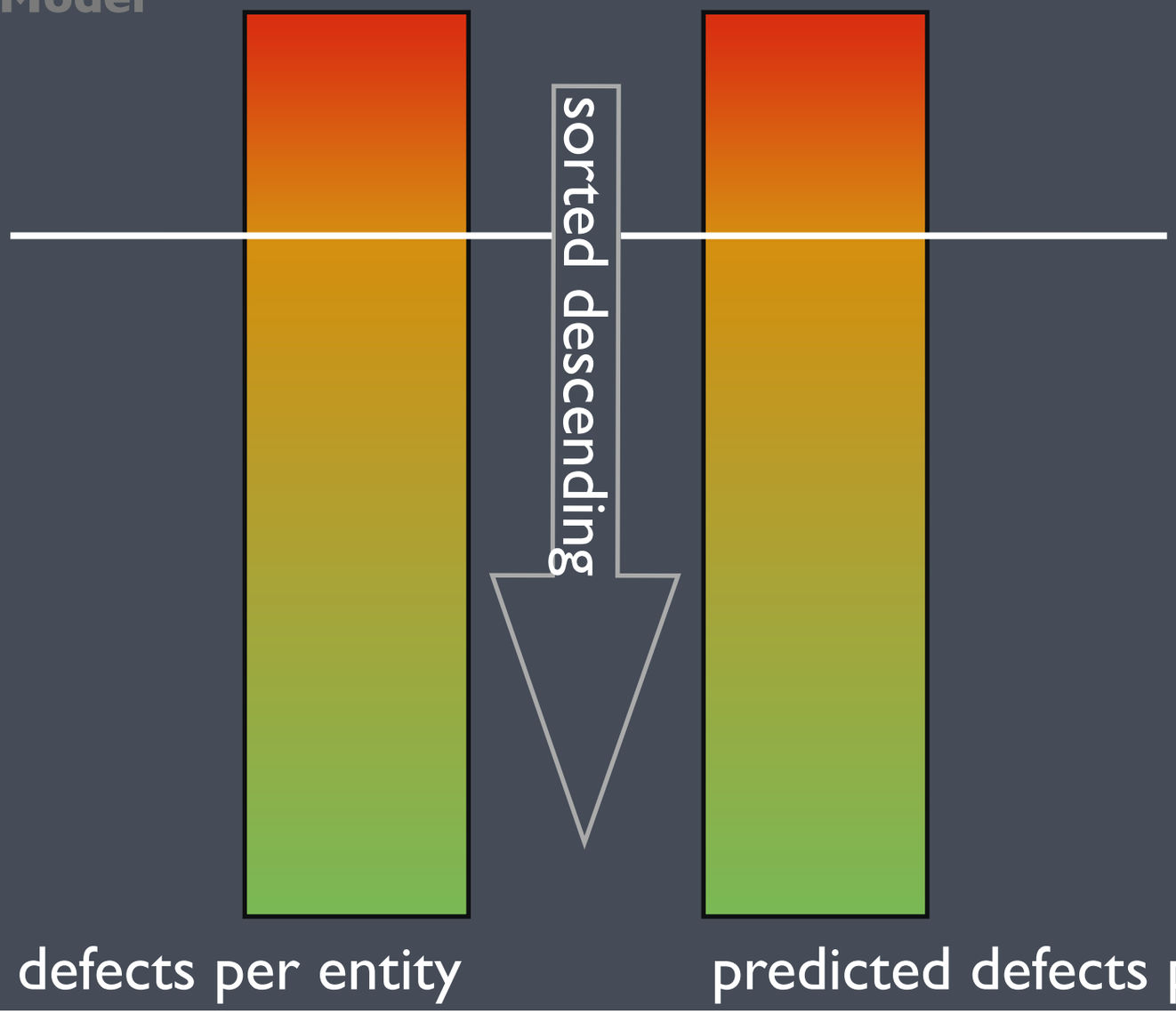
Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

Never predict concrete number!

Because people will take them for real!

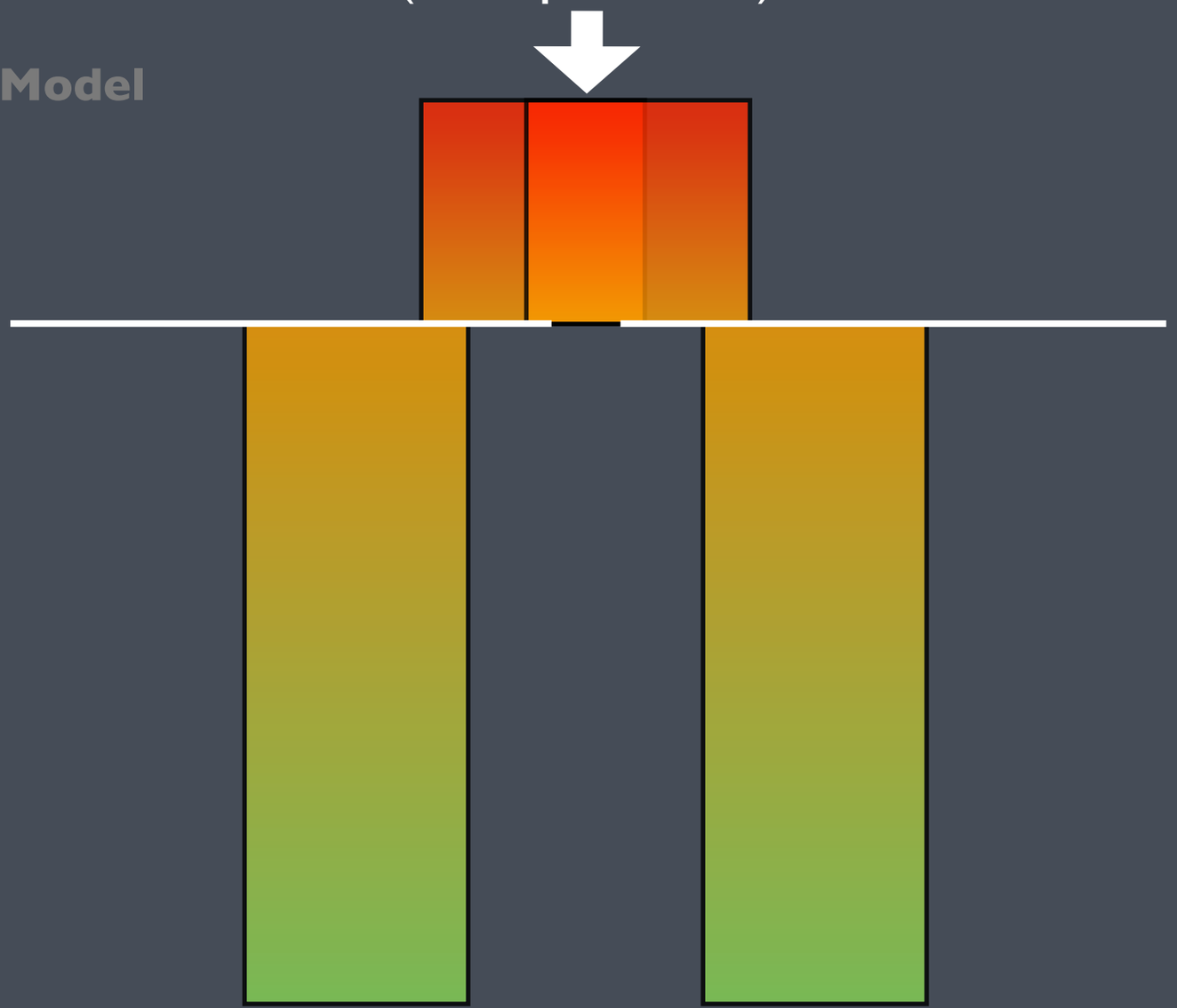
Evaluation

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model



Evaluation

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model



Recall, Precision, Accuracy

		Predict defects ?	
		Yes	No
Real defects?	Yes	**true positives**	**false negatives**
	No	**false positives**	**true negatives**

Recall, Precision, Accuracy

		Predict defects ?	
		Yes	No
Real defects?	Yes	true positives	false negatives
	No	false positives	true negatives

Precision

$$\frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

Predicted defect prone entities will be defect prone!

Recall, Precision, Accuracy

		Predict defects ?	
		Yes	No
Real defects?	Yes	true positives	false negatives
	No	false positives	true negatives

Recall

Precision

$$\frac{\text{true positives}}{\text{true positives} + \text{false negative}}$$

All defect prone entities
get predicted as defect prone.

Recall, Precision, Accuracy

		Predict defects ?	
		Yes	No
Real defects?	Yes	true positives	false negatives
	No	false positives	true negatives

→ **Recall**

↓ **Precision**

↘

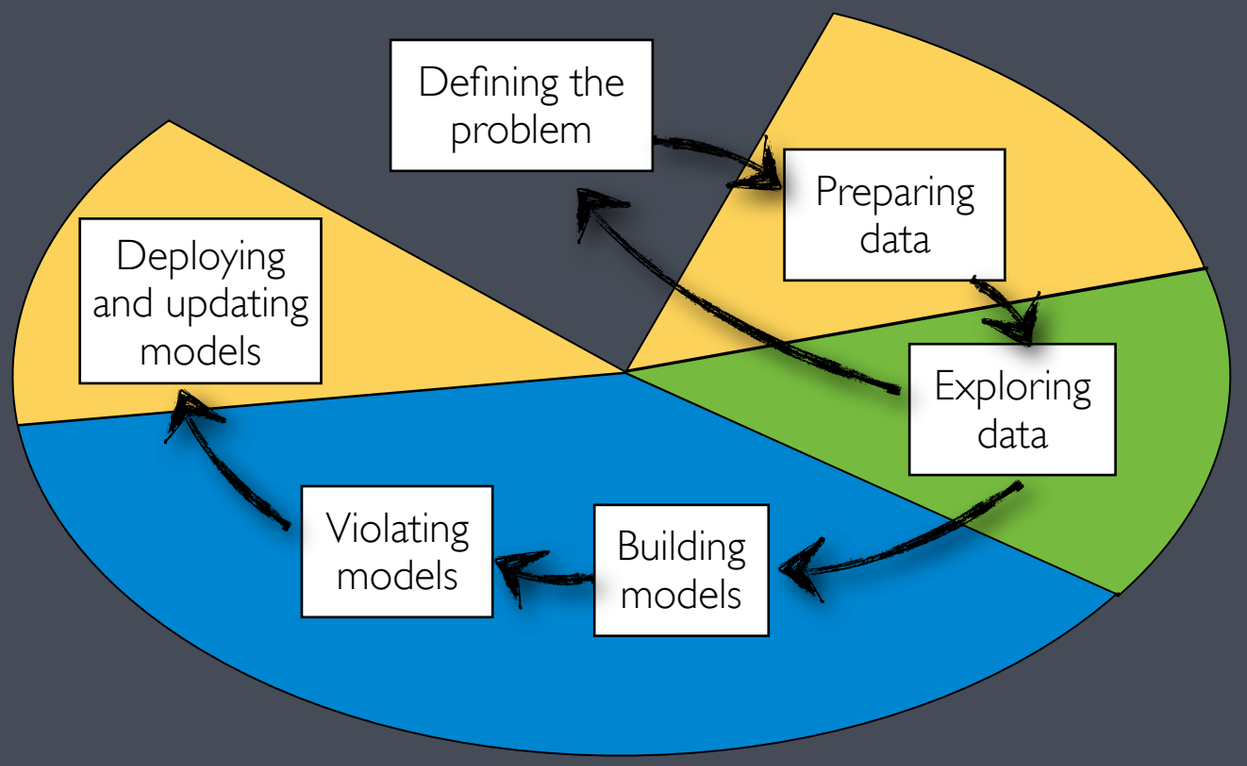
Accuracy

$$\frac{\text{true positives + true negatives}}{\text{true positives + true negatives + false negative + false positives}}$$

The overall correctness

Step 6: Deploying Model

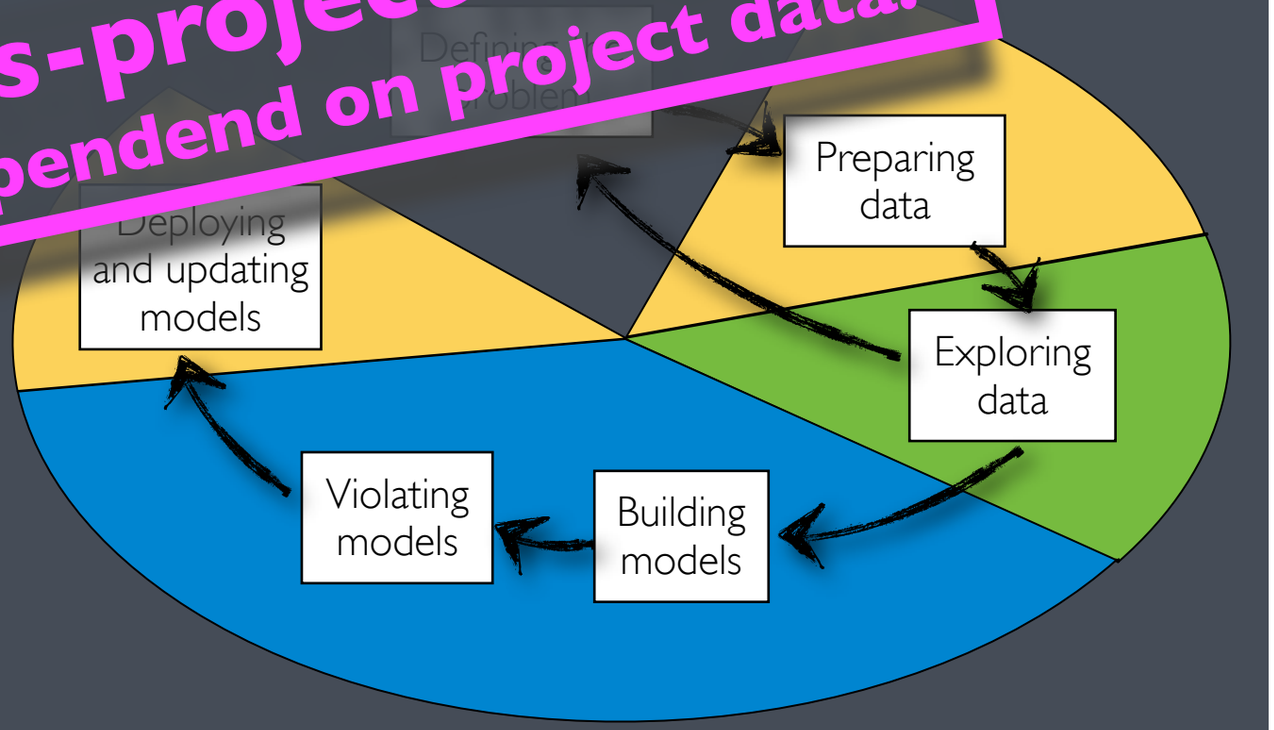
- Integrate model into development or quality assurance process
- Update model frequently
  (because change happens)
- Frequently validate the precision of your model



Step 6: Deploying Model

- Integrate model into development or quality assurance process
- Update model frequently (because change happens)
- Frequently validate the precision of your model

Careful with cross-project models!
Many models highly dependend on project data!



State of the Art

State of the Art

Prediction Results

Training	Testing	Precision	Recall	Accuracy
2.0	2.0	0.692	0.265	0.876
	2.1	0.478	0.191	0.890
	3.0	0.613	0.171	0.861
2.1	2.0	0.664	0.203	0.870
	2.1	0.668	0.160	0.900
	3.0	0.717	0.139	0.864
3.0	2.0	0.578	0.277	0.866
	2.1	0.528	0.220	0.894
	3.0	0.675	0.224	0.869

Predicting Defects for Eclipse
[Revised for Dataset Version 2.0a]

Thomas Zimmermann
Saarland University
tz@acm.org

Rahul Premraj
Saarland University
premraj@cs.uni-sb.de

Andreas Zeller
Saarland University
zeller@acm.org

Predicting java classes: Classification: has bugs, has no bugs

Prediction Results

Training	Testing	Precision	Recall	Accuracy
2.0	2.0	**0.692**	**0.265**	**0.876**
	2.1	**0.478**	**0.191**	**0.890**
	3.0	**0.613**	**0.171**	**0.861**
2.1	2.0	**0.664**	**0.203**	**0.870**
	2.1	**0.668**	**0.160**	**0.900**
	3.0	**0.717**	**0.139**	**0.864**
3.0	2.0	**0.578**	**0.277**	**0.866**
	2.1	**0.528**	**0.220**	**0.894**
	3.0	**0.675**	**0.224**	**0.869**

Predicting Defects for Eclipse
[Revised for Dataset Version 2.0a]

Thomas Zimmermann
Saarland University
tz@acm.org

Rahul Premraj
Saarland University
premraj@cs.uni-sb.de

Andreas Zeller
Saarland University
zeller@acm.org

Predicting java classes: Classification: has bugs, has no bugs

Prediction Results

Training	Testing	Precision	Recall	Accuracy
2.0	2.0	**0.692**	**0.265**	**0.876**
	2.1	**0.478**	**0.191**	**0.890**
	3.0	**0.613**	**0.171**	**0.861**
2.1	2.0	**0.664**	**0.153**	**0.870**
	2.1	**0.658**	**0.160**	**0.900**
	3.0	**0.717**	**0.139**	**0.864**
3.0	2.0	**0.578**	**0.277**	**0.866**
	2.1	**0.528**	**0.220**	**0.894**
	3.0	**0.675**	**0.224**	**0.869**

Complexity causes defects!

eclipse

Predicting Defects for Eclipse
[Revised for Dataset Version 2.0a]

Thomas Zimmermann
Saarland University
tz@acm.org

Rahul Premraj
Saarland University
premraj@cs.uni-sb.de

Andreas Zeller
Saarland University
zeller@acm.org

Predicting java classes: Classification: has bugs, has no bugs

Prediction Results

Training	Testing	Precision	Recall	Accuracy
2.0	2.0	**0.692**	**0.265**	**0.876**
	2.1	**0.478**	**0.191**	**0.890**
	3.0	**0.613**	**0.171**	**0.861**
2.1	2.0	0.664	0.205	0.870
	2.1	0.888	0.180	0.901
	3.0	0.717	0.209	0.864
3.0	2.0	0.588	0.277	0.866
	2.1	0.528	0.220	0.894
	3.0	0.675	0.224	0.869

Complexity causes defects!

But not all defects come from complexity!

eclipse

Predicting Defects for Eclipse
[Revised for Dataset Version 2.0a]

Thomas Zimmermann
Saarland University
tz@acm.org

Rahul Premraj
Saarland University
premraj@cs.uni-sb.de

Andreas Zeller
Saarland University
zeller@acm.org

Predicting java classes: Classification: has bugs, has no bugs

What to mine?

Code

e-mail

Bug Reports

Changes

Profiles

What to mine?

Traces

Effort

Specification

Chats

Tests

Navigation

Models

Code

e-mail

Bug Reports

Changes

Profiles

Traces

Effort

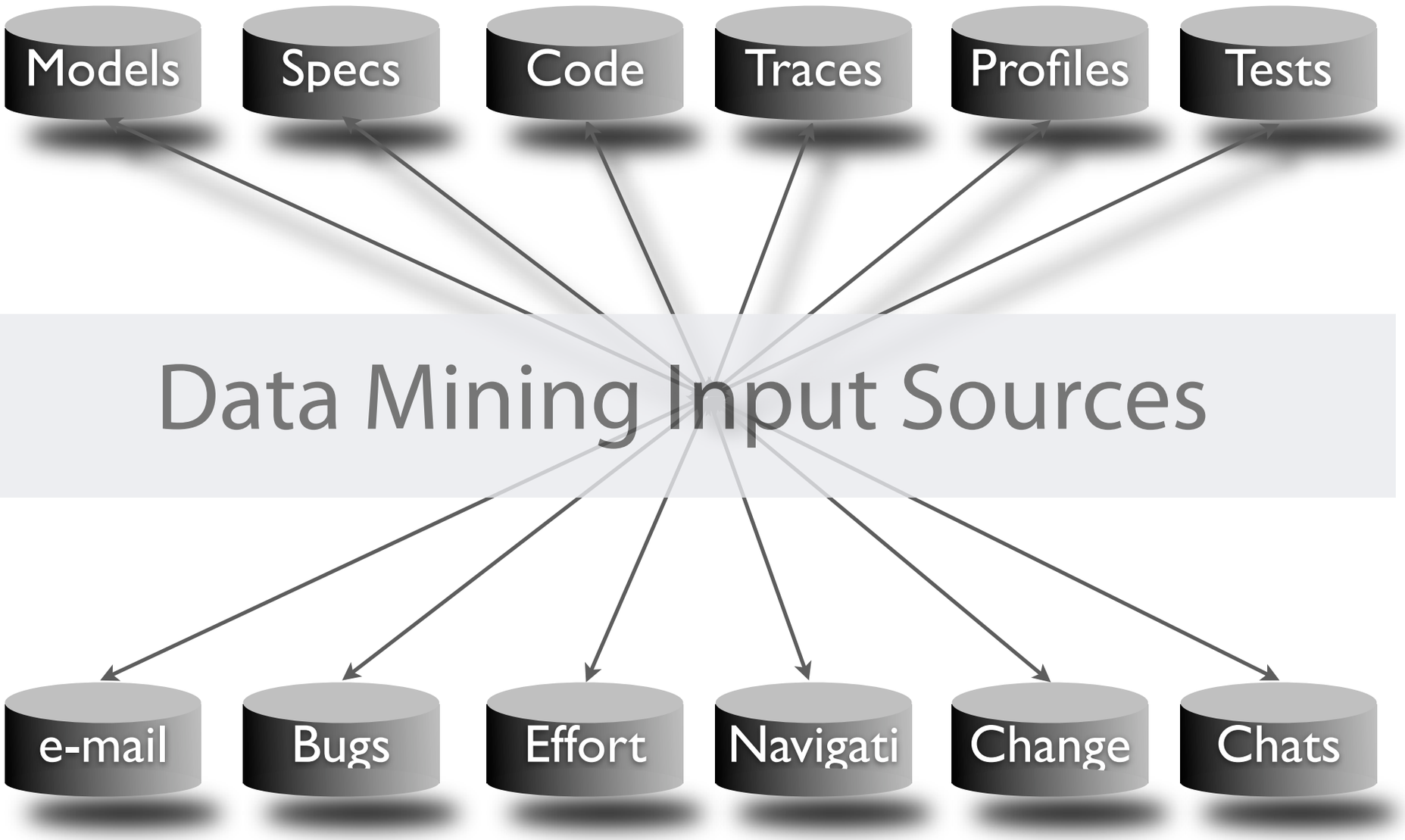
Sepcification

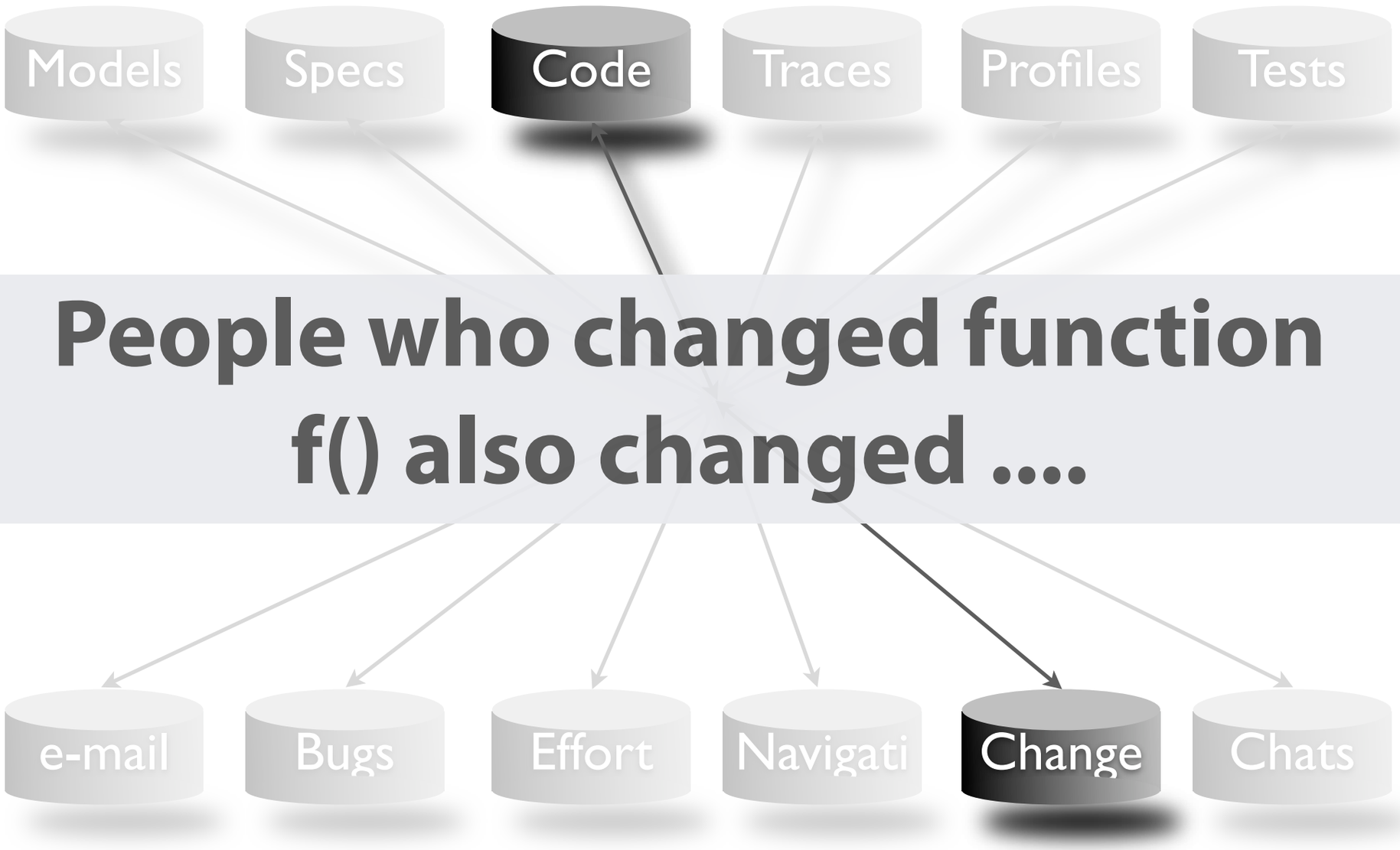
Chats

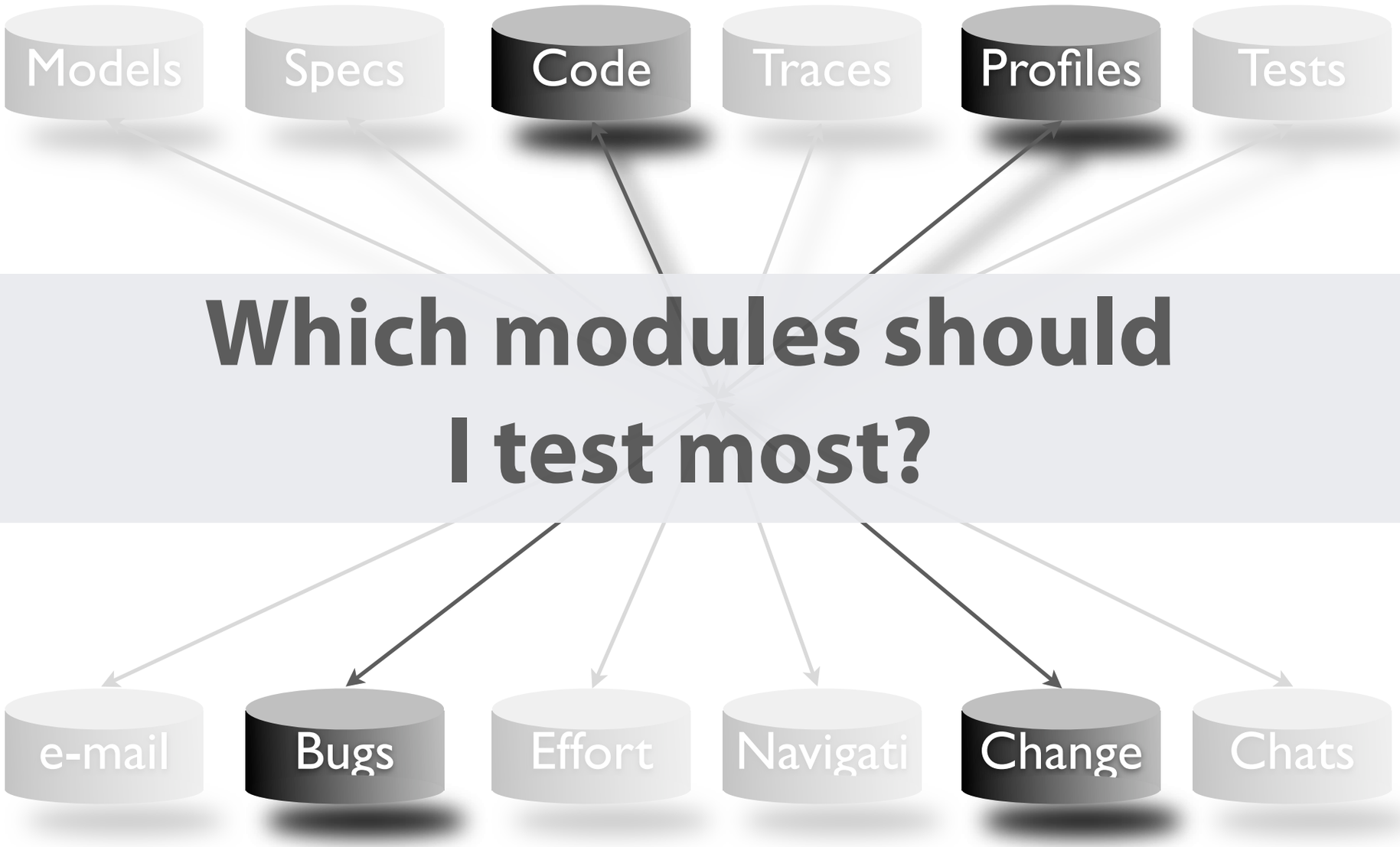
Tests

Navigation

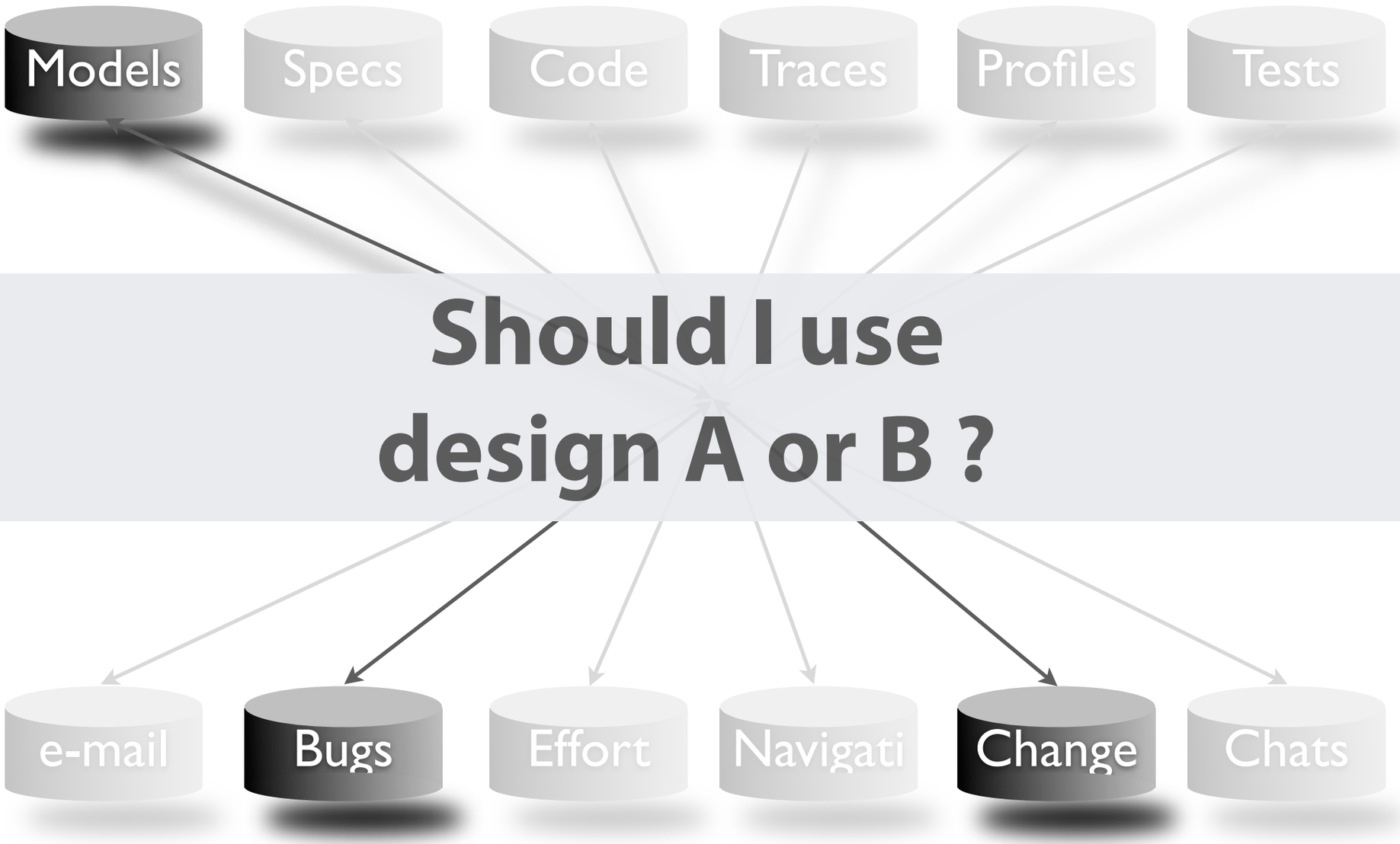
Models

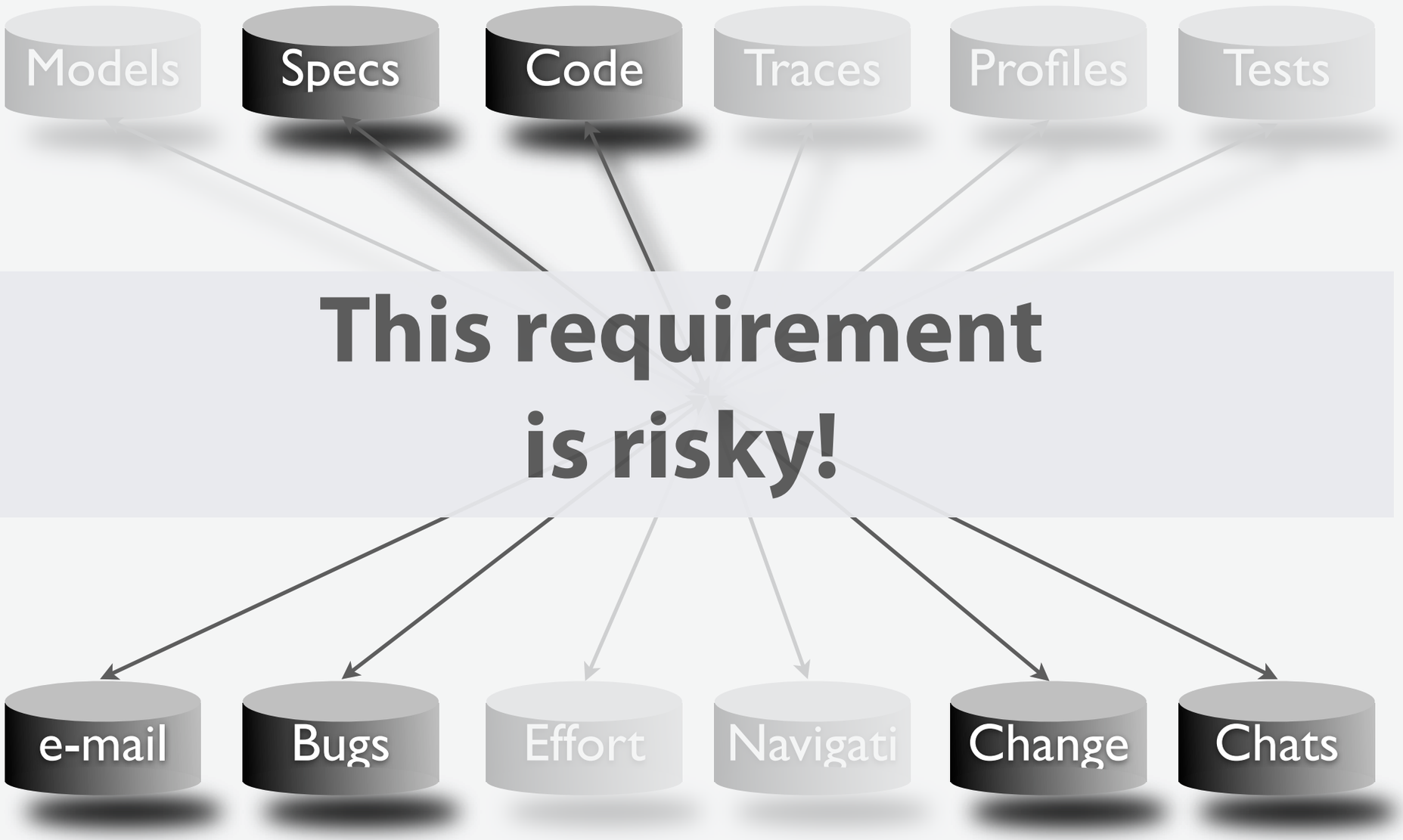

Models
Specs
Code
Traces
Profiles
Tests
Data Mining Input Sources
e-mail
Bugs
Effort
Navigati
Change
Chats


Models
Specs
Code
Traces
Profiles
Tests
People who changed function
f() also changed ....
e-mail
Bugs
Effort
Navigati
Change
Chats

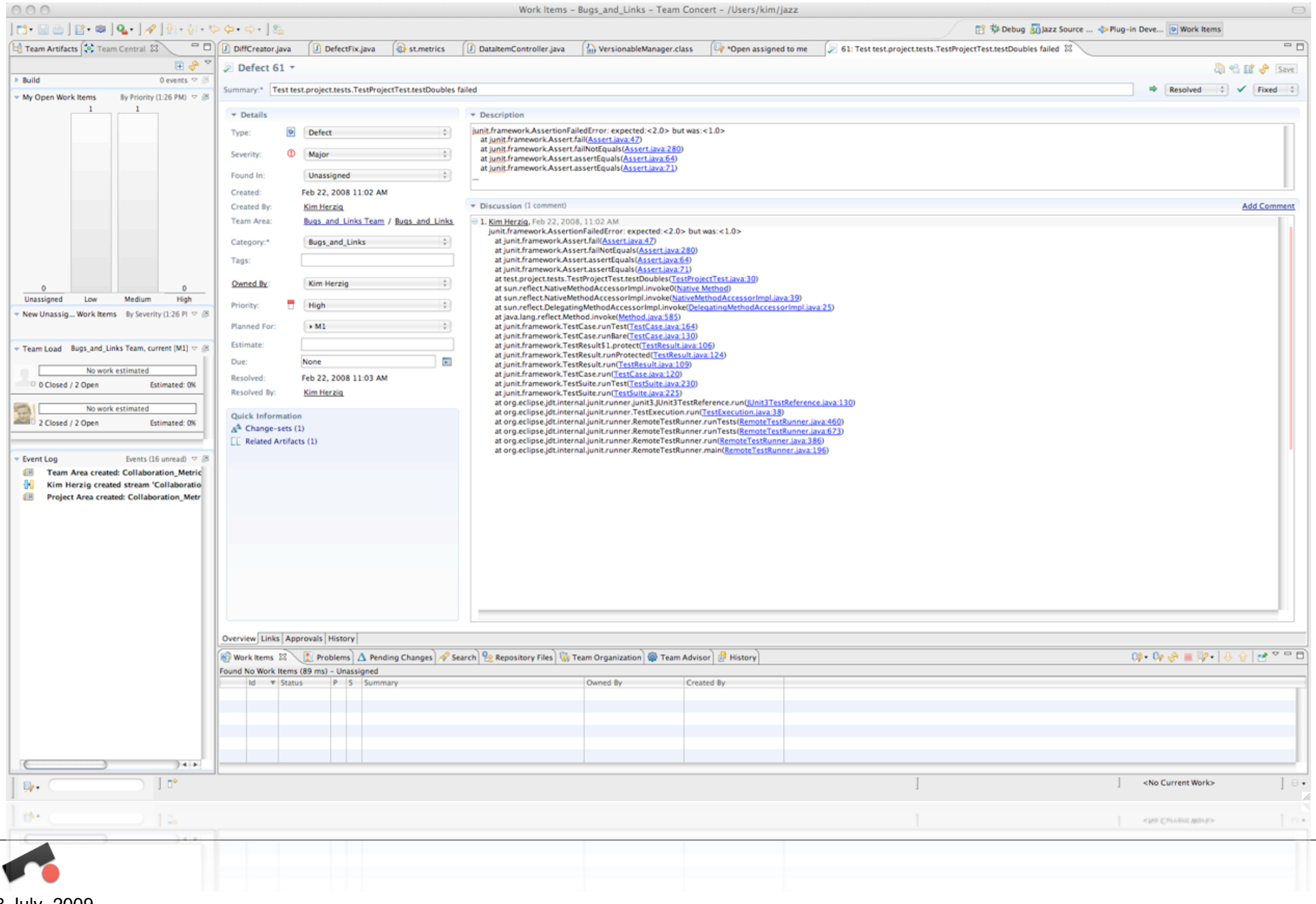

Models
Specs
Code
Traces
Profiles
Tests
Which modules should I test most?
e-mail
Bugs
Effort
Navigati
Change
Chats


Models
Specs
Code
Traces
Profiles
Tests
How long will it take
to fix this bug?
e-mail
Bugs
Effort
Navigati
Change
Chats

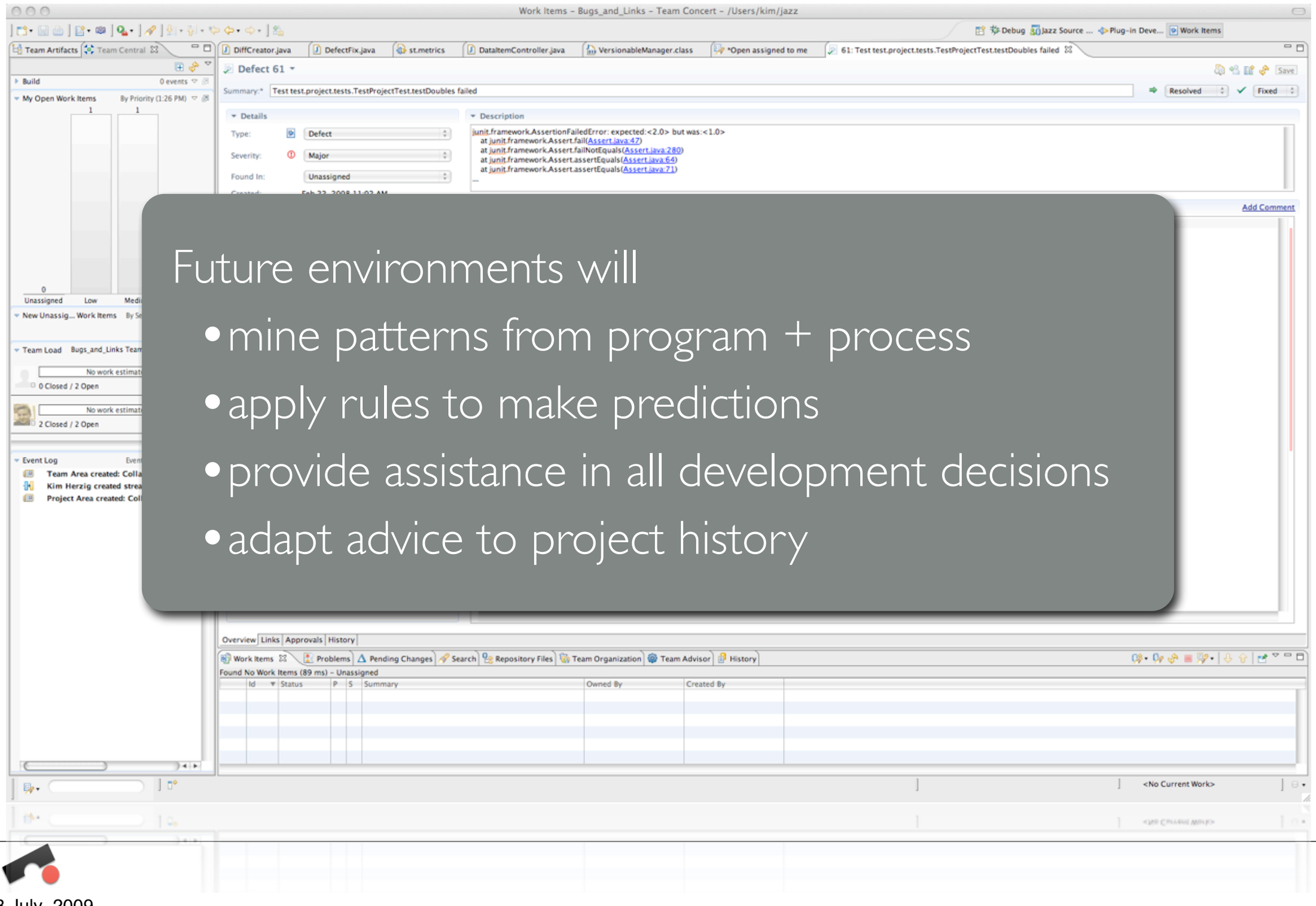

Models
Specs
Code
Traces
Profiles
Tests
Should I use
design A or B ?
e-mail
Bugs
Effort
Navigati
Change
Chats


Models
Specs
Code
Traces
Profiles
Tests
This requirement
is risky!
e-mail
Bugs
Effort
Navigati
Change
Chats

Assistance


Work Items – Bugs_and_Links – Team Concert – /Users/kim/jazz
Team Artifacts
Team Central
Build
0 events
My Open Work Items
By Priority (1:26 PM)
Unassigned
Low
Medium
High
New Unassig... Work Items
By Severity (1:26 PI
Team Load
Bugs_and_Links Team, current [M1]
No work estimated
0 Closed / 2 Open
Estimated: 0%
2 Closed / 2 Open
Event Log
Events (16 unread)
Team Area created: Collaboration_Metric
Kim Herzig created stream 'Collaboratio
Project Area created: Collaboration_Metr
DiffCreator.java
DefectFix.java
st.metrics
DataItemController.java
VersionableManager.class
*Open assigned to me
61: Test test.project.tests.TestProjectTest.testDoubles failed
Defect 61
Save
Summary:*
Test test.project.tests.TestProjectTest.testDoubles failed
Resolved
Fixed
Details
Type:
Defect
Severity:
Major
Found In:
Unassigned
Created:
Feb 22, 2008 11:02 AM
Created By:
Kim Herzig
Team Area:
Bugs_and_Links Team / Bugs_and_Links
Category:*
Bugs_and_Links
Tags:
Owned By:
Kim Herzig
Priority:
High
Planned For:
M1
Estimate:
Due:
None
Resolved:
Feb 22, 2008 11:03 AM
Resolved By:
Kim Herzig
Quick Information
Change-sets (1)
Related Artifacts (1)
Description
junit.framework.AssertionFailedError: expected:<2.0> but was:<1.0>
at junit.framework.Assert.fail(Assert.java:47)
at junit.framework.Assert.failNotEquals(Assert.java:280)
at junit.framework.Assert.assertEquals(Assert.java:64)
at junit.framework.Assert.assertEquals(Assert.java:71)
Discussion (1 comment)
Add Comment
1. Kim Herzig, Feb 22, 2008, 11:02 AM
junit.framework.AssertionFailedError: expected:<2.0> but was:<1.0>
at junit.framework.Assert.fail(Assert.java:47)
at junit.framework.Assert.failNotEquals(Assert.java:280)
at junit.framework.Assert.assertEquals(Assert.java:64)
at junit.framework.Assert.assertEquals(Assert.java:71)
at test.project.tests.TestProjectTest.testDoubles(TestProjectTest.java:30)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:585)
at junit.framework.TestCase.runTest(TestCase.java:164)
at junit.framework.TestCase.runBare(TestCase.java:130)
at junit.framework.TestResult$1.protect(TestResult.java:106)
at junit.framework.TestResult.runProtected(TestResult.java:124)
at junit.framework.TestResult.run(TestResult.java:109)
at junit.framework.TestCase.run(TestCase.java:120)
at junit.framework.TestSuite.runTest(TestSuite.java:230)
at junit.framework.TestSuite.run(TestSuite.java:225)
at org.eclipse.jdt.internal.junit.runner.junit3.JUnit3TestReference.run(JUnit3TestReference.java:130)
at org.eclipse.jdt.internal.junit.runner.TestExecution.run(TestExecution.java:38)
at org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.runTests(RemoteTestRunner.java:460)
at org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.runTests(RemoteTestRunner.java:673)
at org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.run(RemoteTestRunner.java:386)
at org.eclipse.jdt.internal.junit.runner.RemoteTestRunner.main(RemoteTestRunner.java:196)
Overview
Links
Approvals
History
Work Items
Problems
Pending Changes
Search
Repository Files
Team Organization
Team Advisor
History
Found No Work Items (89 ms) – Unassigned
Id
Status
P
S
Summary
Owned By
Created By
<No Current Work>

Assistance


Work Items – Bugs_and_Links – Team Concert – /Users/kim/jazz
Defect 61
Summary:* Test test.project.tests.TestProjectTest.testDoubles failed
Resolved
Fixed
Details
Type: Defect
Severity: Major
Found In: Unassigned
Description
junit.framework.AssertionFailedError: expected:<2.0> but was:<1.0>
at junit.framework.Assert.fail(Assert.java:47)
at junit.framework.Assert.failNotEquals(Assert.java:280)
at junit.framework.Assert.assertEquals(Assert.java:64)
at junit.framework.Assert.assertEquals(Assert.java:71)
Add Comment
Future environments will
• mine patterns from program + process
• apply rules to make predictions
• provide assistance in all development decisions
• adapt advice to project history
Overview
Links
Approvals
History
Work Items
Problems
Pending Changes
Search
Repository Files
Team Organization
Team Advisor
History
Found No Work Items (89 ms) – Unassigned
Id
Status
P
S
Summary
Owned By
Created By
<No Current Work>

Empirical SE 2.0

Wikis

Joy of Use

Participation

Usability

Recommendation

Social Software

Simplicity

Collaboration

Perpetual Beta

Empirical SE 2.0

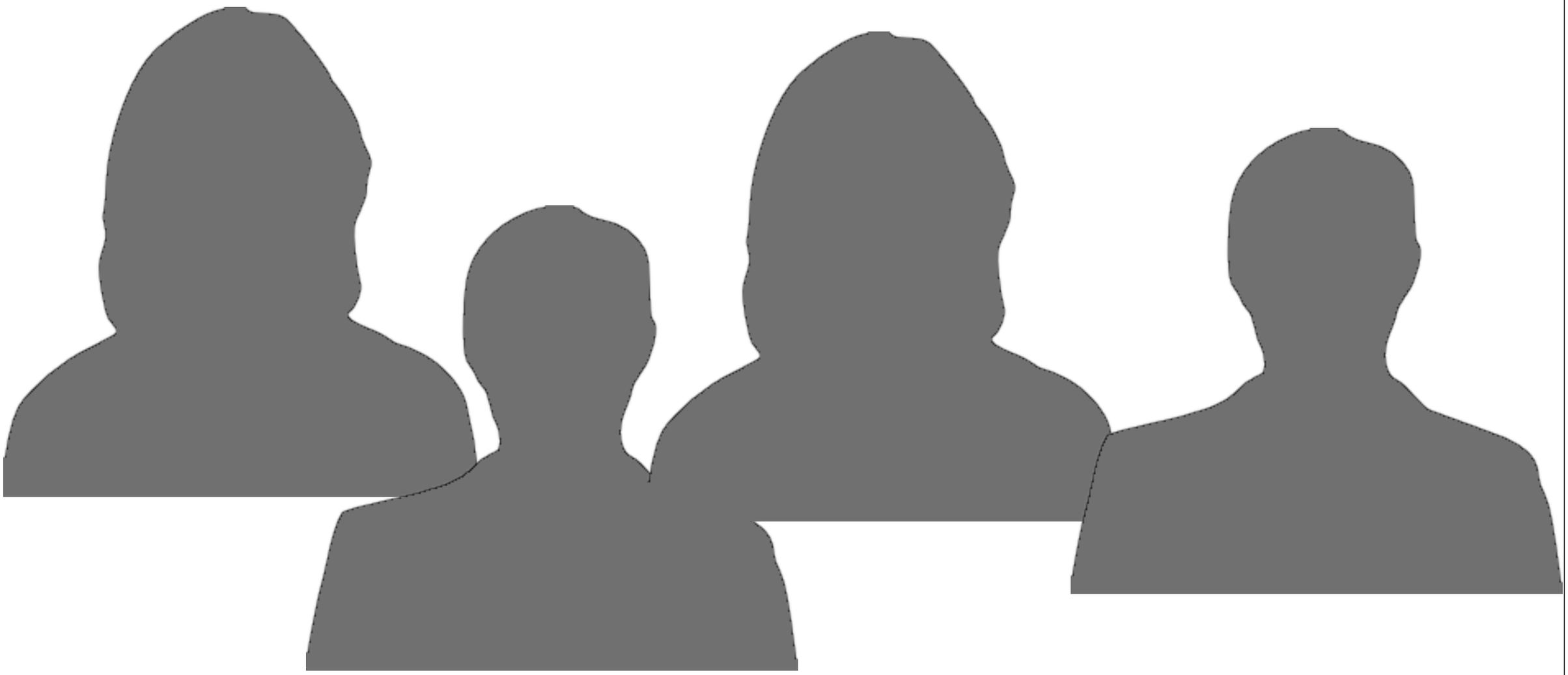
Trust

Economy

Remixability

The Long Tail

DataDriven

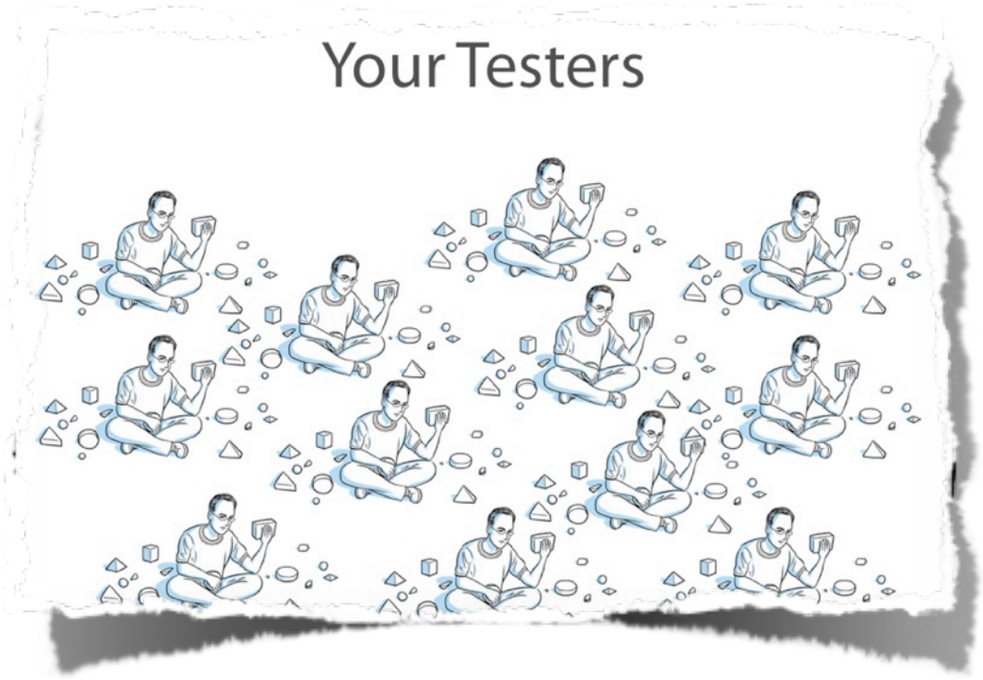


Bachelor/Master Theses in software mining

SAARLAND UNIVERSITY
COMPUTER SCIENCE

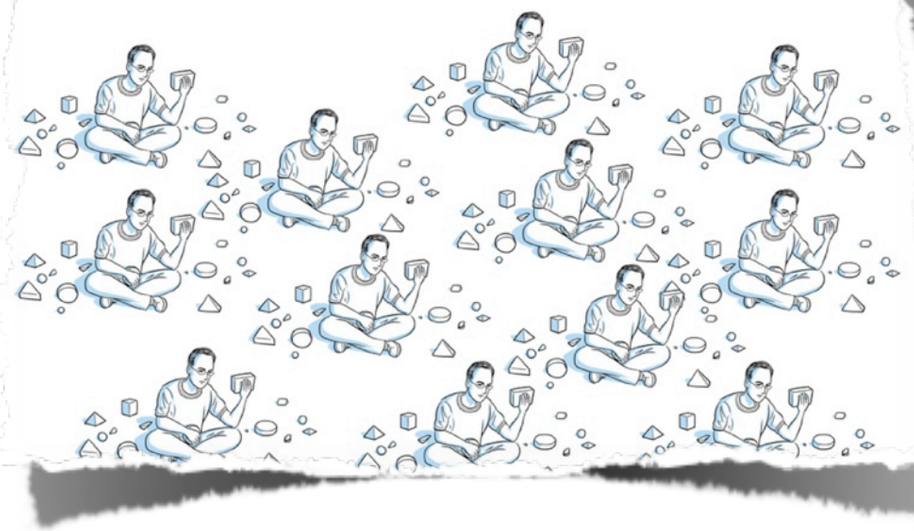
Summary

Summary

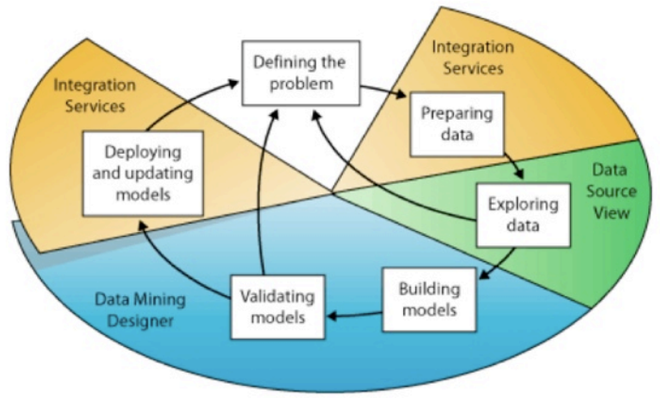


Summary

Your Testers



Building a mining model

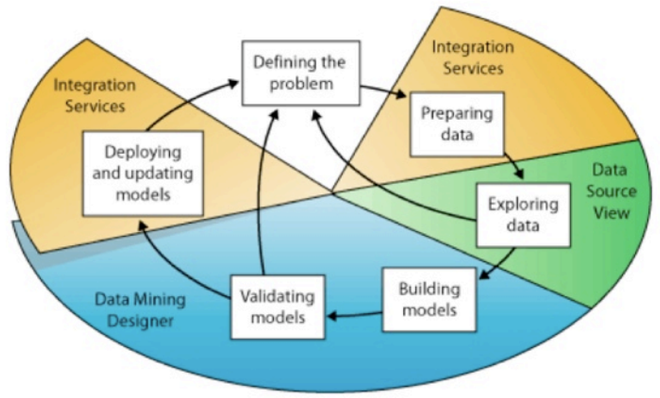


Source of image: http://technet.microsoft.com/en-us/library/ms174949.aspx

Summary

Your Testers

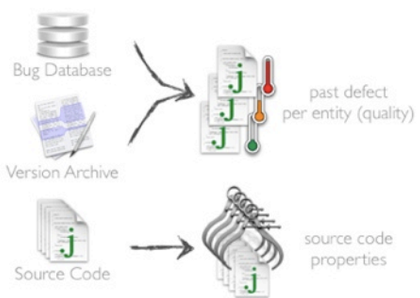
Building a mining model



Source of image: http://technet.microsoft.com/en-us/library/ms174949.aspx

Building the Model

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model

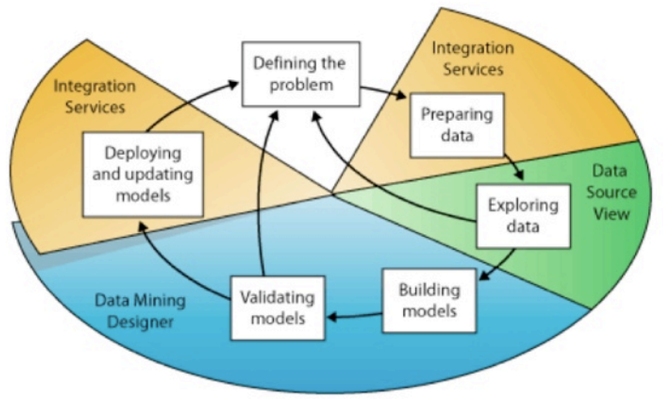


Do defects and metrics correlate?

Summary

Your Testers

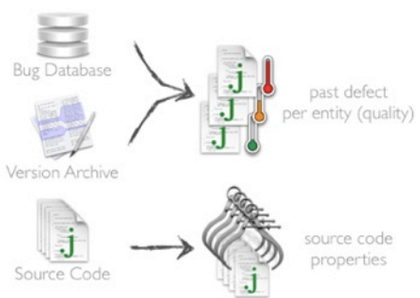
Building a mining model



Source of image: http://technet.microsoft.com/en-us/library/ms174949.aspx

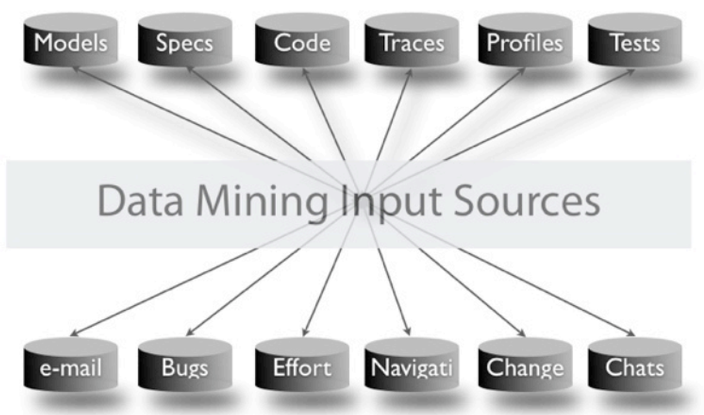
Building the Model

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model



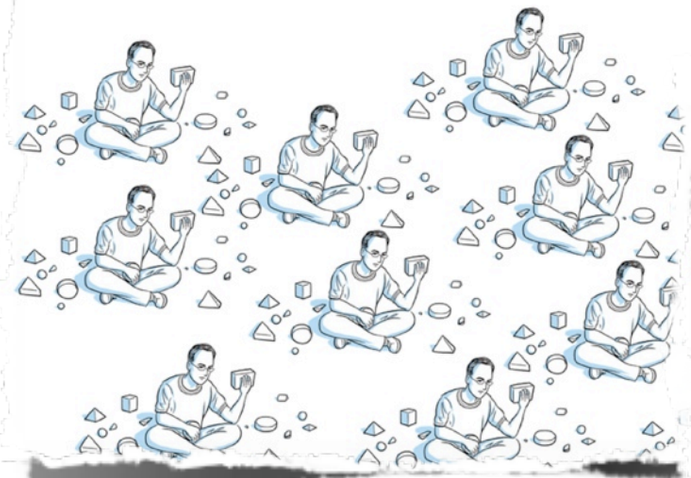
Do defects and metrics correlate?

Data Mining Input Sources



Summary

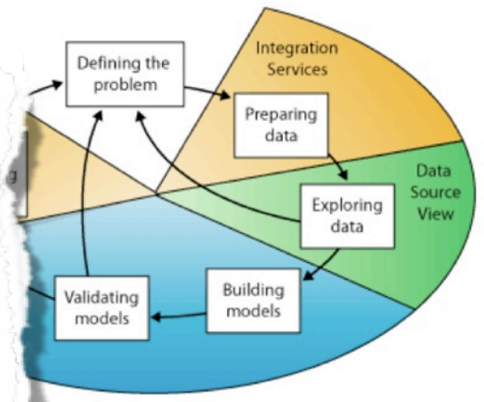
Your Testers



Empirical SE 2.0

Wikis
Joy of Use
Participation
Usability
Recommendation
Social Software
Collaboration
Perpetual Beta
Simplicity
Trust
Economy
The Long Tail
Remixability
DataDriven

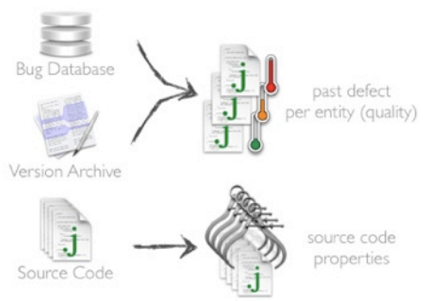
Building a mining model



technet.microsoft.com/en-us/library/ms174949.aspx

Building the Mod

Step 1: Define the problem
Step 2: Prepare Data
Step 3: Explore Data
Step 4: Building the Model
Step 5: Validating the Model



Do defects and metrics correlate?

Data Mining Input Sources