# Integrating Refactoring Recommendation into an IDE: A JetBrains Story

Timofey Bryksin

International Workshop on Refactoring, 2021

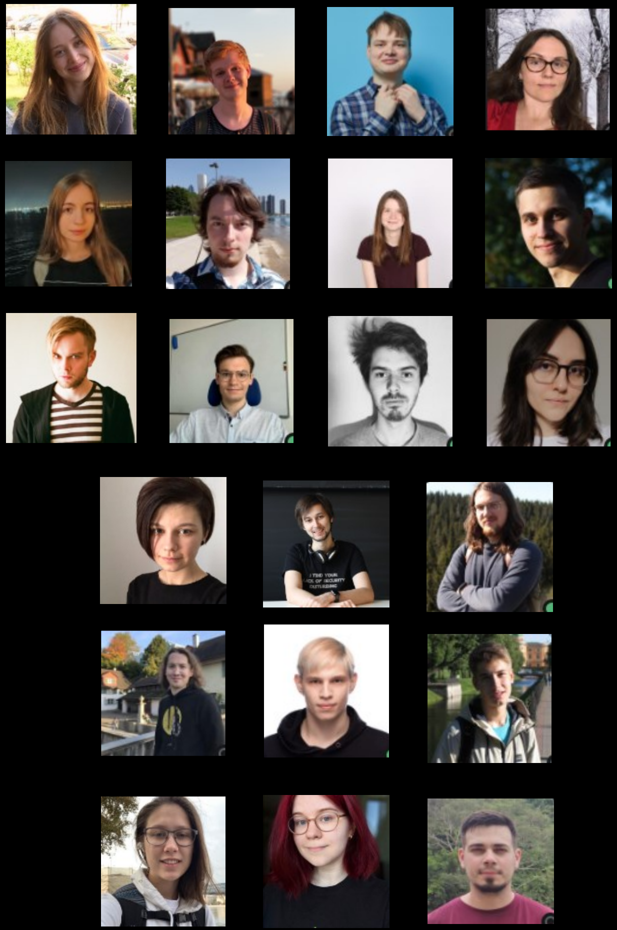
# About JetBrains

- 10+ million users
- 99 companies from the Fortune Top 100 are clients
- 30 products
  - IDEs
  - tools for team work
  - Kotlin
- 1500+ employees in 9 offices around the world
- 18 research labs

https://www.jetbrains.com/lp/annualreport-2020/

# ML4SE Lab

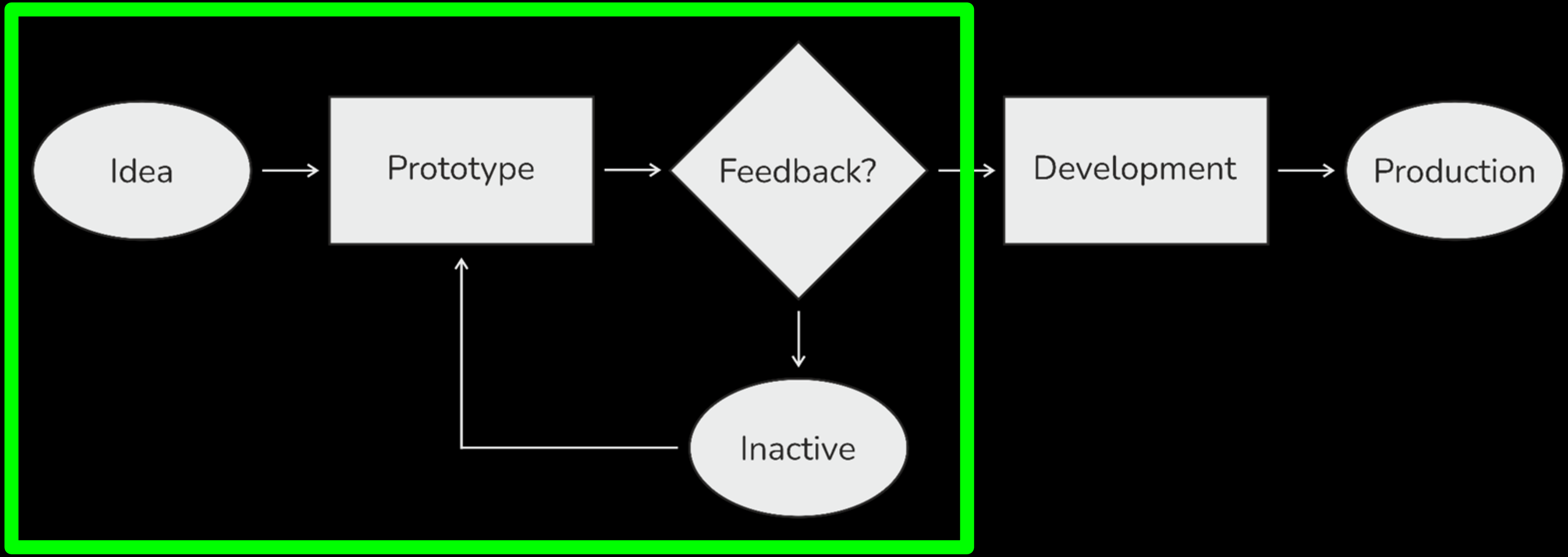
- Founded in Spring 2017
- Data-driven SE
  - we help computers leverage data
    to help people program other computers
- 21 researchers
  - + almost 20 interns from various universities

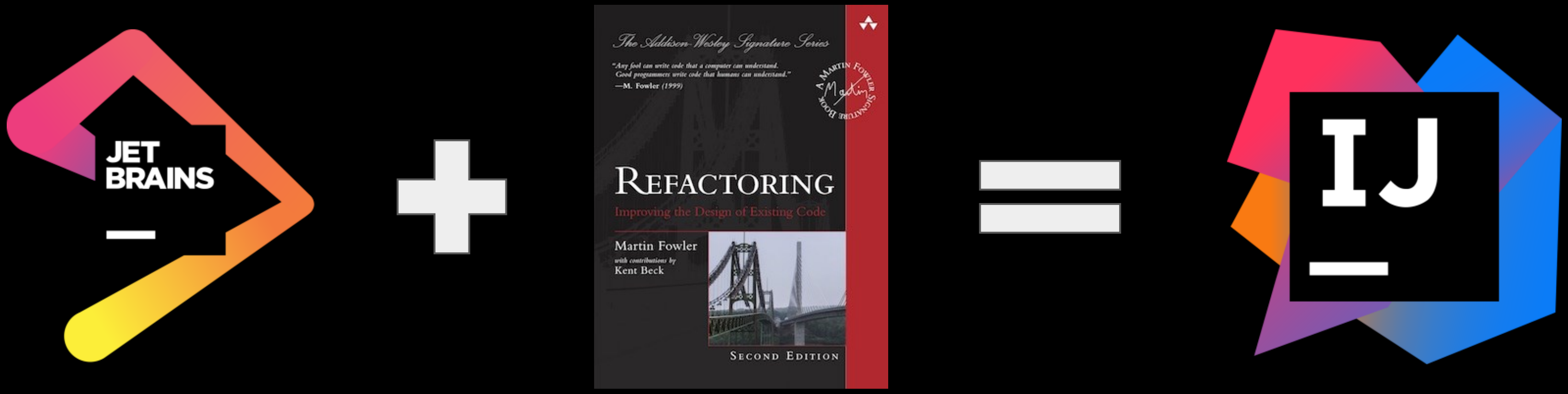


# What this talk is about

- Three stories of refactoring-related IDE features
  - motivation
  - design and implementation details
  - challenges of adoption
  - takeaways for the research community

# Typical Feature Pipeline





*The Addison-Wesley Signature Series*

*"Any fool can write code that a computer can understand. Good programmers write code that humans can understand."*
—M. Fowler *(1999)*

A Martin Fowler Signature Book

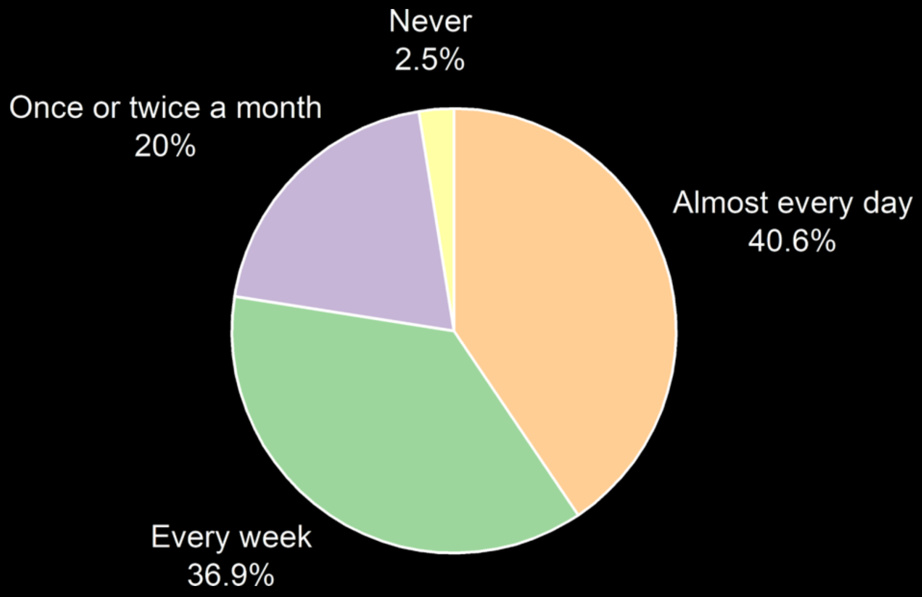
Martin

REFACTORING

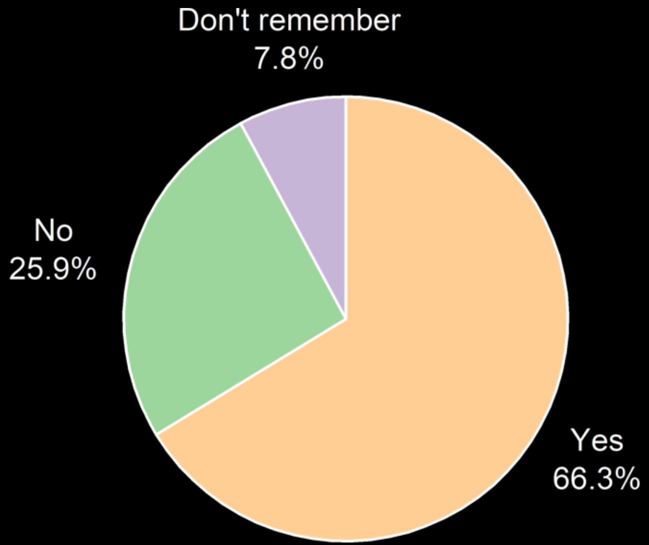
Improving the Design of Existing Code

Martin Fowler
*with contributions by*
Kent Beck

SECOND EDITION



*In the past month, how often have you performed any code refactoring?* (Out of 1,181 respondents)



*During this time, did you ever refactor code for an hour or more in a single session?* (Out of 1,145 respondents)

*Golubev et al.* One Thousand and One Stories: A Large-Scale Survey of Software Refactoring, ESEC/FSE 2021

# Refactoring Recommendation

### Identifying Refactoring Opportunities in Object-Oriented Code: A Systematic Literature Review[1]

Jehad Al Dallal
*Department of Information Science*
*Kuwait University*

**Abstract**
Context: Identi
precedes the ac
to identify oppo
Objective: Thi
opportunities fo
Method: We p
relevant studies
and selected 47
analyzed based
refactoring opp
used.
Results: The re
highly active.
nonindustrial d
considered refa
approaches to
identification te
which helps to
sets used in the
Conclusions: It
activities, invol

**Keywords**: ref

### JMove: A novel heuristic and tool to detect move method refactoring opportunities

Ricardo Terra[a,*], Marco Tulio Valente[b], Sergio Miranda[b], Vitor Sales[b]

[a] Department of Computer Science, Federal University of Lavras, Lavras, Brazil

### Automated Refactoring using Design Differencing

Iman Hemati Moghadam
*School of Computer Science and Informatics*
*University College Dublin, Ireland*
*Email: Iman.Hemati-Moghadam@ucdconnect.ie*

Mel Ó Cinnéide
*School of Computer Science and Informatics*
*University College Dublin, Ireland*
*Email: mel.ocinneide@ucd.ie*

IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 35, NO. 3, MAY/JUNE 2009 347

### Identification of Move Method Refactoring Opportunities

Nikolaos Tsantalis, *Student Member*, *IEEE*, and Alexander Chatzigeorgiou, *Member*, *IEEE*

**Abstract**—P
aided by app
nontrivial to i
Method refac
the notion of
based on the
Therefore, ou
how well ent
since the des
quality criteri
efficiency asp

**Index Terms**

1 INTRODUCT

ACCORDING t
oriented de
strive for low c
empirical studies
and cohesion me
et al. [3] and Br
metrics can ser
Briand et al. [8]
positive correlat
effects, changeal

### Identification of generalization refactoring opportunities

**Hui Liu · Zhendong Niu · Zhiyi Ma ·**
**Weizhong Shao**

Received: 25 July 2011 / Accep
© Springer Science+Business M

**Abstract** Generalization
ing both interfaces and im

### A robust multi-objective approach to balance severity and importance of refactoring opportunities

**Mohamed Wiem Mkaouer[1] · Marouane Kessentini[1] ·**
**Mel Ó Cinnéide[2] · Shinpei Hayashi[3] · Kalyanmoy Deb[4]**

v York 2016

ns involves several sources of uncertainty related to the
corrected and the importance of the classes in which the

### A Review on Search-Based Tools and Techniques to Identify Bad Code Smells in Object-Oriented Systems

**Amandeep Kaur and Gaurav Dhiman**

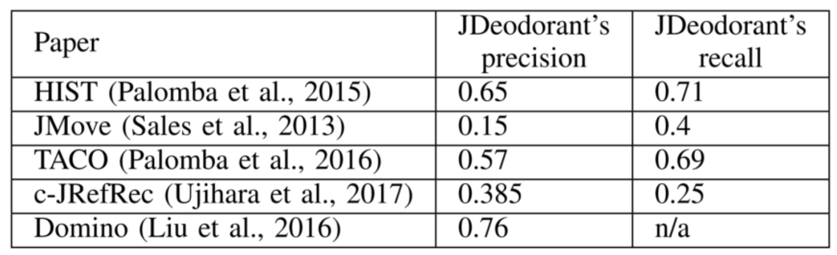
# Story #1: ArchitectureReloaded (2017-2018)

# The Plan

- Find the best recommendation algorithm
- Build an IntelliJ IDEA plugin around
- See how it works
- …
- PROFIT!

# Types of Evaluation We Faced

- Case studies on small projects where all refactorings are obvious
- Expert assessment of the algorithm’s result on a real-world project
- Tracking software metrics
- Evaluation on refactorings mined from historical data
- Evaluation on a labeled dataset
- Evaluation on a dataset with artificially introduced code smells

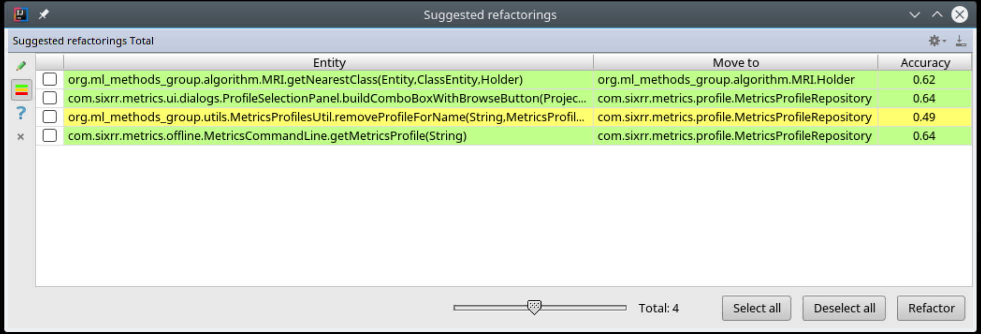


| Paper | JDeodorant’s precision | JDeodorant’s recall |
|---|---|---|
| HIST (Palomba et al., 2015) | 0.65 | 0.71 |
| JMove (Sales et al., 2013) | 0.15 | 0.4 |
| TACO (Palomba et al., 2016) | 0.57 | 0.69 |
| c-JRefRec (Ujihara et al., 2017) | 0.385 | 0.25 |
| Domino (Liu et al., 2016) | 0.76 | n/a |

*Tsantalis et al.* Ten Years of JDeodorant: Lessons Learned from the Hunt for Smells (2018)

# ArchitectureReloaded

- Targeting the Move Method refactoring
- Implemented three different ML-based approaches
  - community detection
  - clustering in a metric-based vector space
- Tons of implementation tweeks
- Applied ensemble/voting to get better results



Suggested refactorings Total

|  | Entity | Move to | Accuracy |
|---|---|---|---|
| □ | org.ml_methods_group.algorithm.MRI.getNearestClass(Entity,ClassEntity,Holder) | org.ml_methods_group.algorithm.MRI.Holder | 0.62 |
| □ | com.sixrr.metrics.ui.dialogs.ProfileSelectionPanel.buildComboBoxWithBrowseButton(Projec... | com.sixrr.metrics.profile.MetricsProfileRepository | 0.64 |
| □ | org.ml_methods_group.utils.MetricsProfilesUtil.removeProfileForName(String,MetricsProfil... | com.sixrr.metrics.profile.MetricsProfileRepository | 0.49 |
| □ | com.sixrr.metrics.offline.MetricsCommandLine.getMetricsProfile(String) | com.sixrr.metrics.profile.MetricsProfileRepository | 0.64 |

Total: 4

# Takeaways

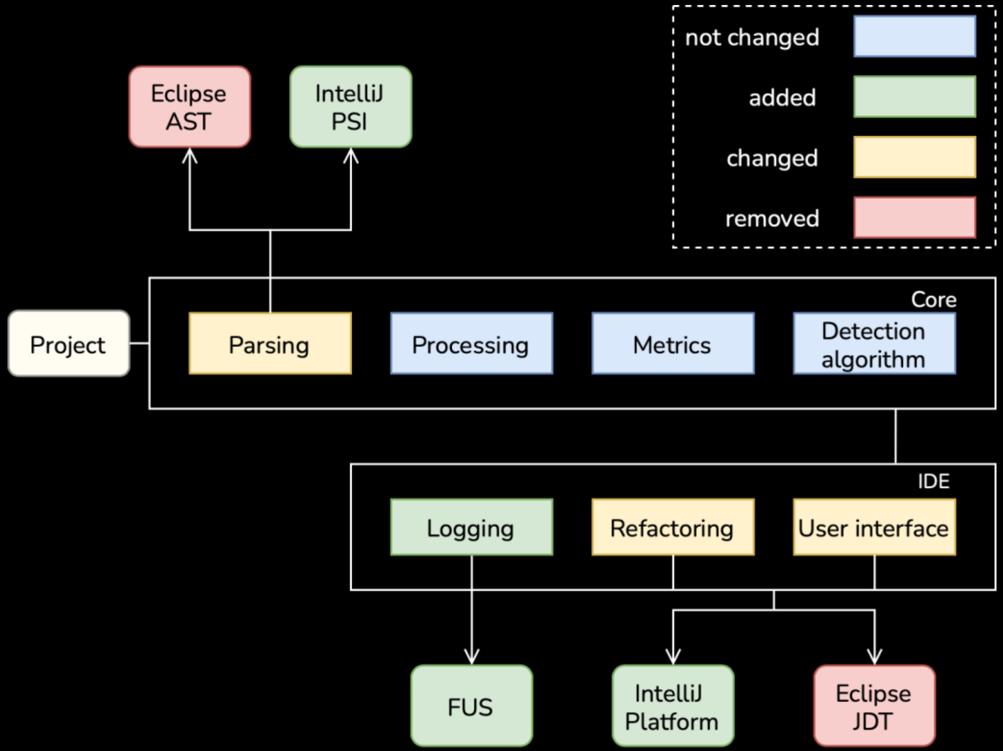
- Providing an open-source replication package is essential
  - 10 pages are almost never enough to describe everything
- A good benchmark is half of the solution
  - invest in a comparison platform
  - collect datasets for different code smells/refactoring types
- Refactoring recommendations vs Hints for improvement
  - chains of refactoring operations
  - integration with IDE is key

# Story #2: IntelliJDeodorant (2019-2020)

# JDeodorant

- Feature Envy, Long Method, Type/State Checking, God Class, Duplicated code
- High precision and recall
- Based on Eclipse JDT

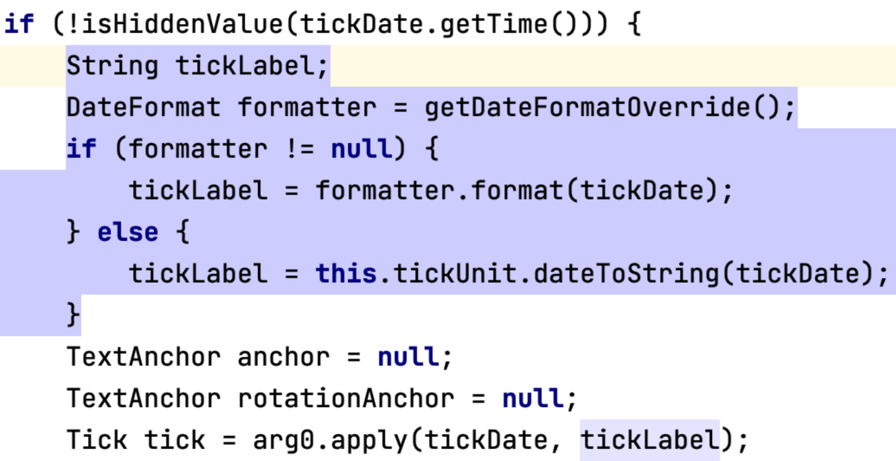
# JDeodorant → IntelliJDeodorant



# Collection of User Logs

- Based on the FUS (Feature Usage Statistics) infrastructure
- Saving description on the code instead of the code itself
- Everything we collect is completely anonymous

# Example of an Extract Method Refactoring



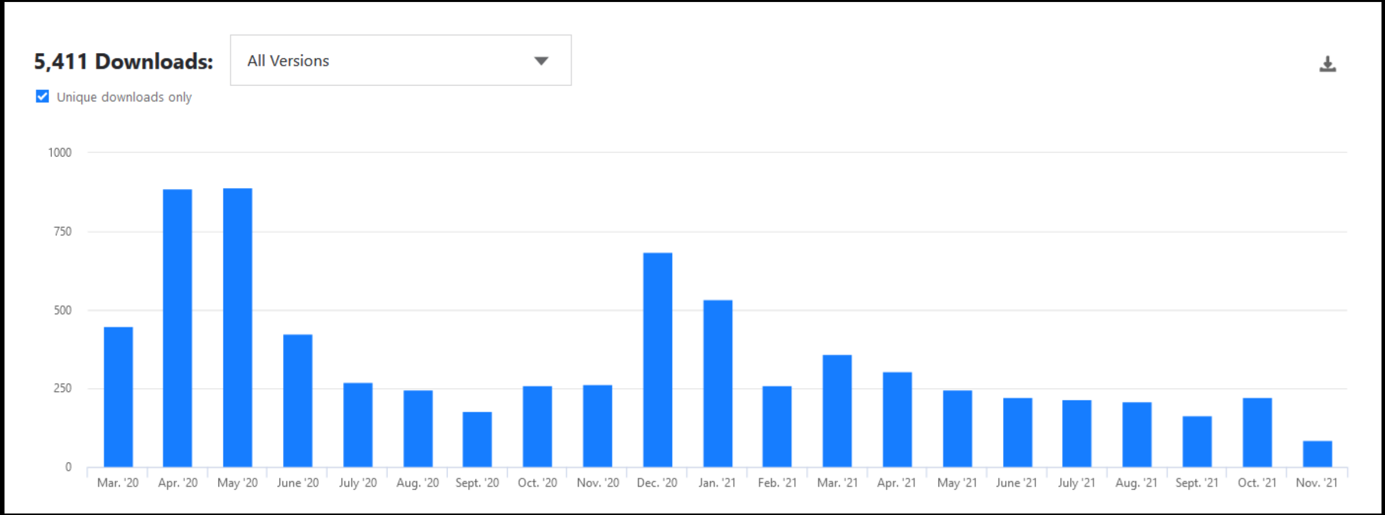
```
if (!isHiddenValue(tickDate.getTime())) {
    String tickLabel;
    DateFormat formatter = getDateFormatOverride();
    if (formatter != null) {
        tickLabel = formatter.format(tickDate);
    } else {
        tickLabel = this.tickUnit.dateToString(tickDate);
    }
    TextAnchor anchor = null;
    TextAnchor rotationAnchor = null;
    Tick tick = arg0.apply(tickDate, tickLabel);
```

What we get:

extracted_statements_count = 5
new_method_length = 8
new_method_parameters_count = 1
original_method_length_before = 53
original_method_length_after = 47
original_method_parameters_count = 4

https://blog.jetbrains.com/author/roman-poborchiy-jetbrains-com/

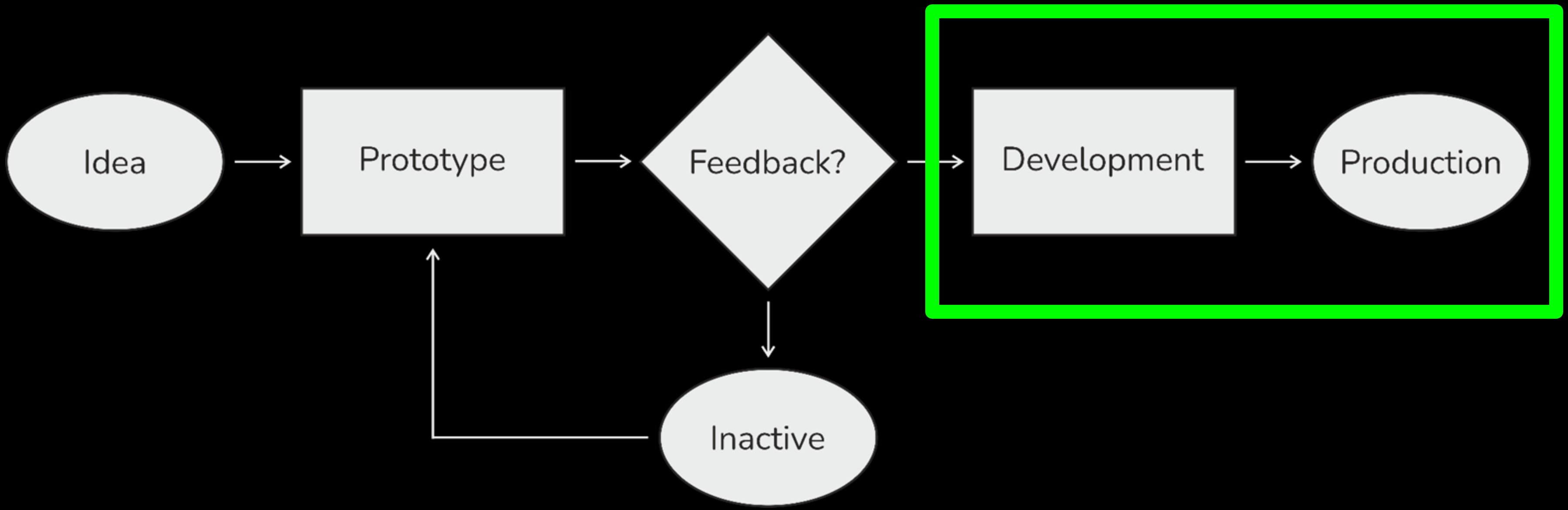
# The IntelliJDeodorant Plugin



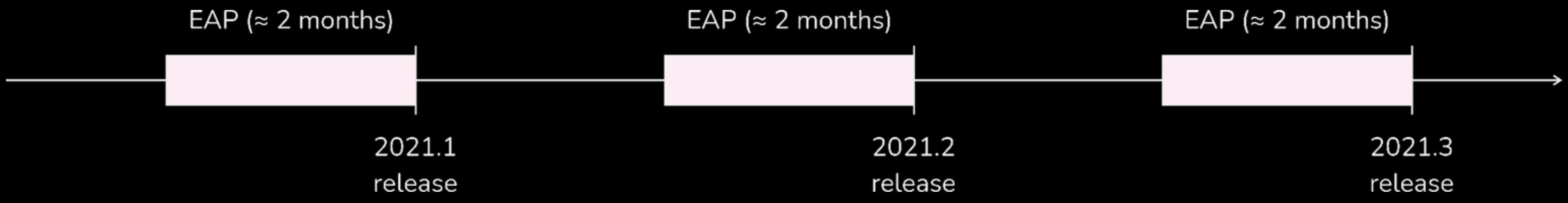
**5,411 Downloads:** All Versions

☑ Unique downloads only

# Back to the Pipeline



# Early Access Program



EAP (≈ 2 months)

EAP (≈ 2 months)

EAP (≈ 2 months)

2021.1 release

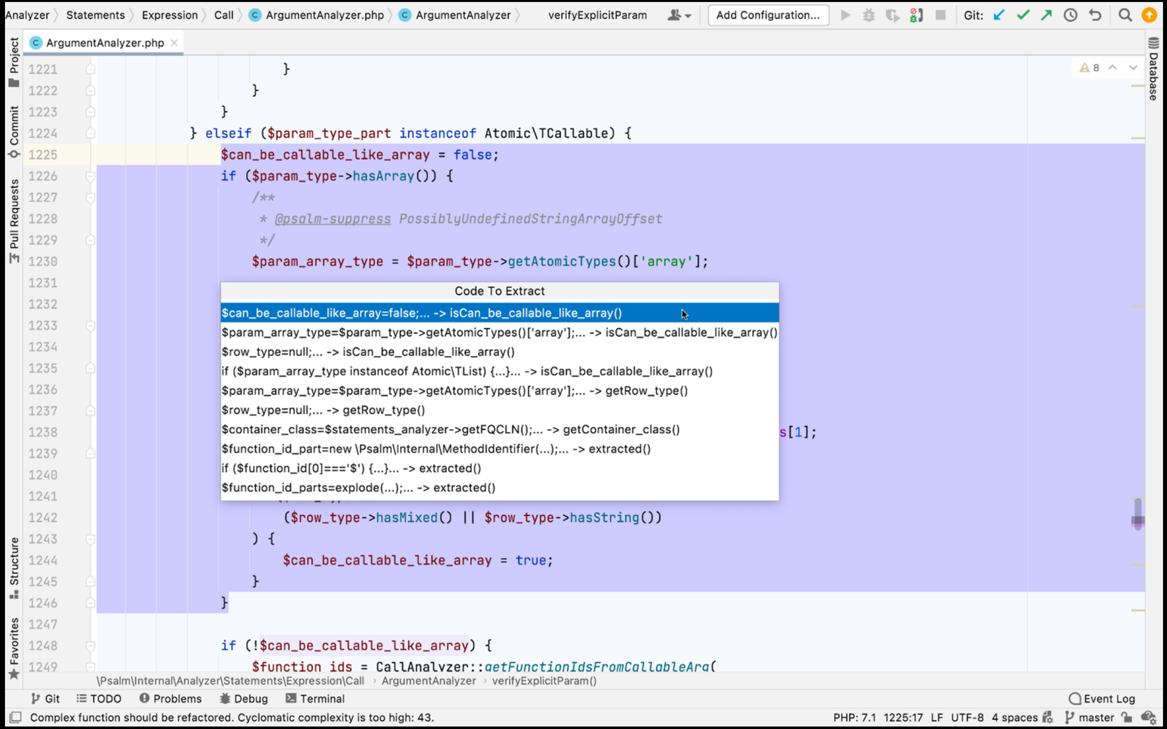
2021.2 release

2021.3 release

https://www.jetbrains.com/resources/eap/

# PhpStorm 2021.2 EAP

- Showing several candidates in a popup window



Analyzer 〉 Statements 〉 Expression 〉 Call 〉 ArgumentAnalyzer.php 〉 ArgumentAnalyzer 〉 verifyExplicitParam
Add Configuration...
Git:
ArgumentAnalyzer.php
Project
Commit
Pull Requests
Structure
Favorites
Database
⚠ 8

```php
                }
            }
        }
    } elseif ($param_type_part instanceof Atomic\TCallable) {
        $can_be_callable_like_array = false;
        if ($param_type->hasArray()) {
            /**
             * @psalm-suppress PossiblyUndefinedStringArrayOffset
             */
            $param_array_type = $param_type->getAtomicTypes()['array'];
```

Code To Extract

```
$can_be_callable_like_array=false;... -> isCan_be_callable_like_array()
$param_array_type=$param_type->getAtomicTypes()['array'];... -> isCan_be_callable_like_array()
$row_type=null;... -> isCan_be_callable_like_array()
if ($param_array_type instanceof Atomic\TList) {...}... -> isCan_be_callable_like_array()
$param_array_type=$param_type->getAtomicTypes()['array'];... -> getRow_type()
$row_type=null;... -> getRow_type()
$container_class=$statements_analyzer->getFQCLN();... -> getContainer_class()
$function_id_part=new \Psalm\Internal\MethodIdentifier(...);... -> extracted()
if ($function_id[0]==='$') {...}... -> extracted()
$function_id_parts=explode(...);... -> extracted()
```

```php
s[1];
                ($row_type->hasMixed() || $row_type->hasString())
            ) {
                $can_be_callable_like_array = true;
            }
        }

        if (!$can_be_callable_like_array) {
            $function_ids = CallAnalyzer::getFunctionIdsFromCallableArg(
```

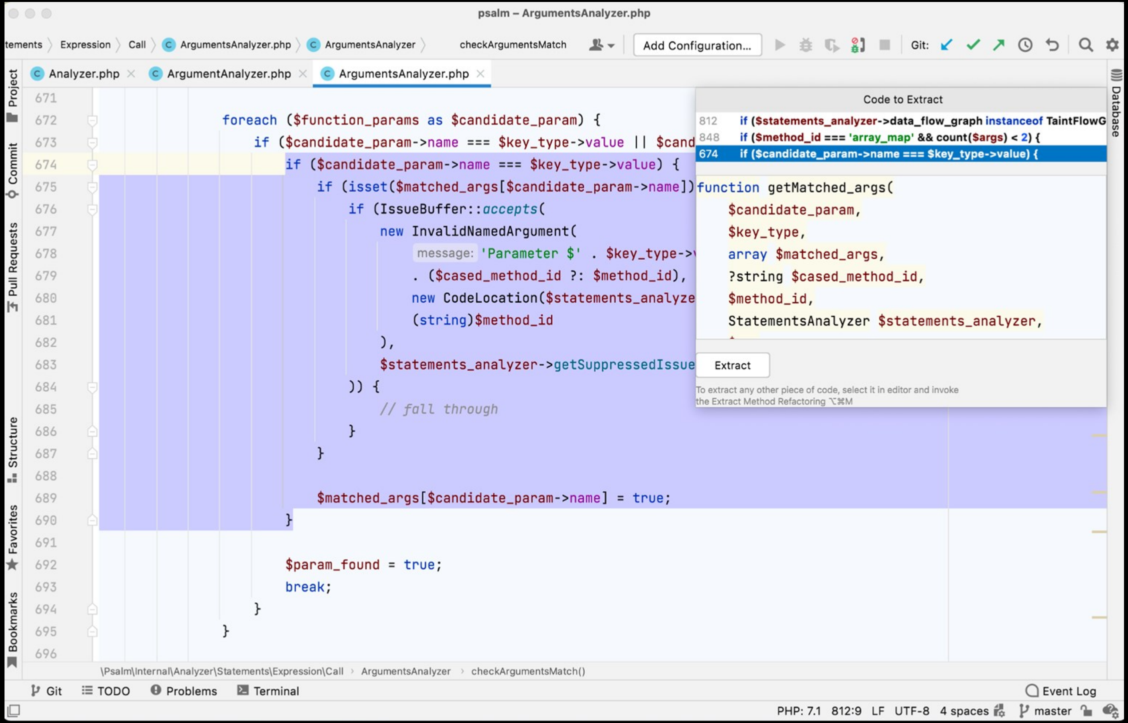
\Psalm\Internal\Analyzer\Statements\Expression\Call › ArgumentAnalyzer › verifyExplicitParam()
Git
TODO
Problems
Debug
Terminal
Event Log

Complex function should be refactored. Cyclomatic complexity is too high: 43.

*Tsantalis and Chatzigeorgiou*. Identification of extract method refactoring opportunities for the decomposition of methods (JSS, 2011)

# PhpStorm 2021.3 EAP

- Showing several best candidates
- Improved UX


psalm – ArgumentsAnalyzer.php
tements 〉 Expression 〉 Call 〉 ArgumentsAnalyzer.php 〉 ArgumentsAnalyzer 〉 checkArgumentsMatch
Add Configuration...
Git:
Analyzer.php
ArgumentAnalyzer.php
ArgumentsAnalyzer.php
Project
Commit
Pull Requests
Structure
Favorites
Bookmarks
Database
\Psalm\Internal\Analyzer\Statements\Expression\Call › ArgumentsAnalyzer › checkArgumentsMatch()
Git
TODO
Problems
Terminal
Event Log
PHP: 7.1
812:9
LF
UTF-8
4 spaces
master

```
foreach ($function_params as $candidate_param) {
    if ($candidate_param->name === $key_type->value || $cand
        if ($candidate_param->name === $key_type->value) {
            if (isset($matched_args[$candidate_param->name])
                if (IssueBuffer::accepts(
                    new InvalidNamedArgument(
                        message: 'Parameter $' . $key_type->
                        . ($cased_method_id ?: $method_id),
                        new CodeLocation($statements_analyze
                        (string)$method_id
                    ),
                    $statements_analyzer->getSuppressedIssue
                )) {
                    // fall through
                }
            }

            $matched_args[$candidate_param->name] = true;
        }

        $param_found = true;
        break;
    }
}
```

Code to Extract

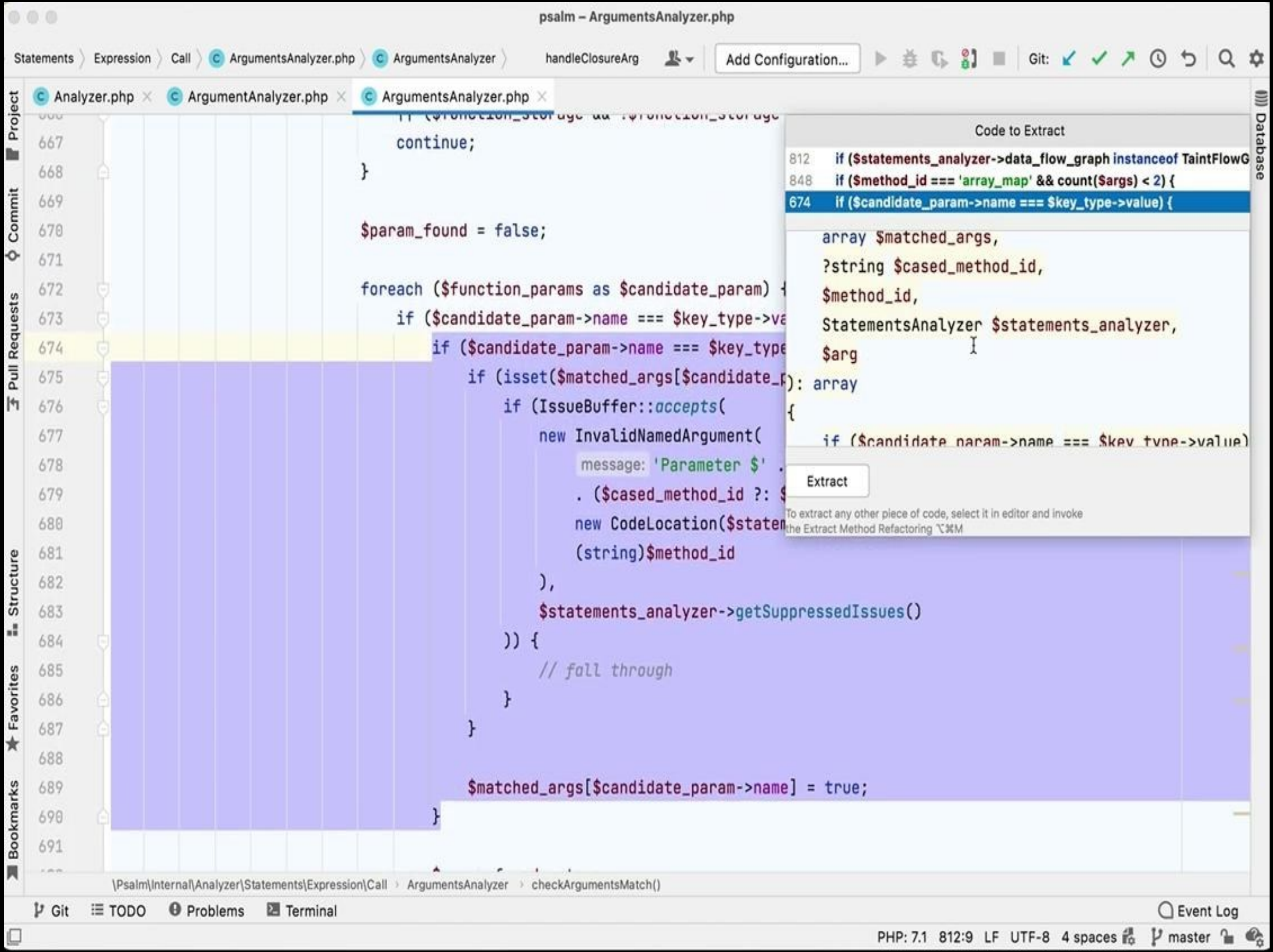
```
812    if ($statements_analyzer->data_flow_graph instanceof TaintFlowG
848    if ($method_id === 'array_map' && count($args) < 2) {
674    if ($candidate_param->name === $key_type->value) {
```

```
function getMatched_args(
    $candidate_param,
    $key_type,
    array $matched_args,
    ?string $cased_method_id,
    $method_id,
    StatementsAnalyzer $statements_analyzer,
```

Extract

To extract any other piece of code, select it in editor and invoke the Extract Method Refactoring ⌥⌘M

*Haas and Hummel*. Deriving Extract Method Refactoring Suggestions for Long Methods (SWQD, 2016)



```
if ($function_storage && !$function_storage
    continue;
}

$param_found = false;

foreach ($function_params as $candidate_param) {
    if ($candidate_param->name === $key_type->va
        if ($candidate_param->name === $key_type
            if (isset($matched_args[$candidate_p
                if (IssueBuffer::accepts(
                    new InvalidNamedArgument(
                        message: 'Parameter $' .
                        . ($cased_method_id ?: $
                        new CodeLocation($statem
                        (string)$method_id
                    ),
                    $statements_analyzer->getSuppressedIssues()
                )) {
                    // fall through
                }
            }

            $matched_args[$candidate_param->name] = true;
        }
```

Code to Extract

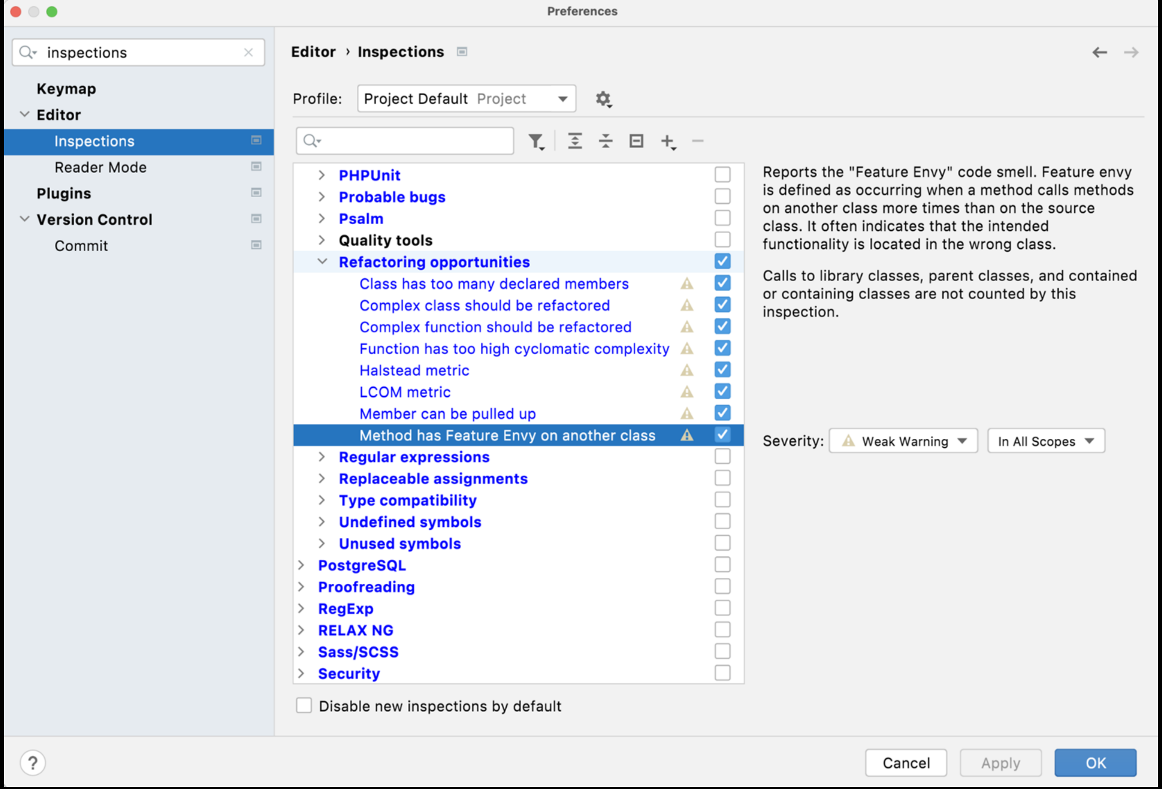
```
812  if ($statements_analyzer->data_flow_graph instanceof TaintFlowG
848  if ($method_id === 'array_map' && count($args) < 2) {
674  if ($candidate_param->name === $key_type->value) {
```

```
    array $matched_args,
    ?string $cased_method_id,
    $method_id,
    StatementsAnalyzer $statements_analyzer,
    $arg
): array
{
    if ($candidate_param->name === $key_type->value)
```

To extract any other piece of code, select it in editor and invoke the Extract Method Refactoring ⌥⌘M

\Psalm\Internal\Analyzer\Statements\Expression\Call › ArgumentsAnalyzer › checkArgumentsMatch()
Git
TODO
Problems
Terminal
Event Log
PHP: 7.1
812:9
LF
UTF-8
4 spaces
master

# PhpStorm 2021.3 EAP



Preferences

inspections

- Keymap
- Editor
  - Inspections
  - Reader Mode
- Plugins
- Version Control
  - Commit

Editor › Inspections

Profile: Project Default Project

- PHPUnit □
- Probable bugs □
- Psalm □
- Quality tools □
- Refactoring opportunities ☑
  - Class has too many declared members ⚠ ☑
  - Complex class should be refactored ⚠ ☑
  - Complex function should be refactored ⚠ ☑
  - Function has too high cyclomatic complexity ⚠ ☑
  - Halstead metric ⚠ ☑
  - LCOM metric ⚠ ☑
  - Member can be pulled up ⚠ ☑
  - Method has Feature Envy on another class ⚠ ☑
- Regular expressions □
- Replaceable assignments □
- Type compatibility □
- Undefined symbols □
- Unused symbols □
- PostgreSQL □
- Proofreading □
- RegExp □
- RELAX NG □
- Sass/SCSS □
- Security □

Reports the "Feature Envy" code smell. Feature envy is defined as occurring when a method calls methods on another class more times than on the source class. It often indicates that the intended functionality is located in the wrong class.

Calls to library classes, parent classes, and contained or containing classes are not counted by this inspection.

Severity: ⚠ Weak Warning In All Scopes

□ Disable new inspections by default

# Takeaways

- What do developers actually want from the refactoring recommendation tool?
  - identify the places where refactoring is needed indeed
  - show only a couple of the best suggestions (maybe just even one)
- We should think not only about *what* to suggest but also *how*
  - refactoring tools should not break the flow
  - are the current tools implemented in the best way possible?
    - Gail Murphy’s ICSME’21 Keynote
- Performance is as important as precision
  - filtering out unsuitable candidates as early as possible
  - use the data pre-calculated by the IDE

# Story #3: RefactorInsight (2020-...)

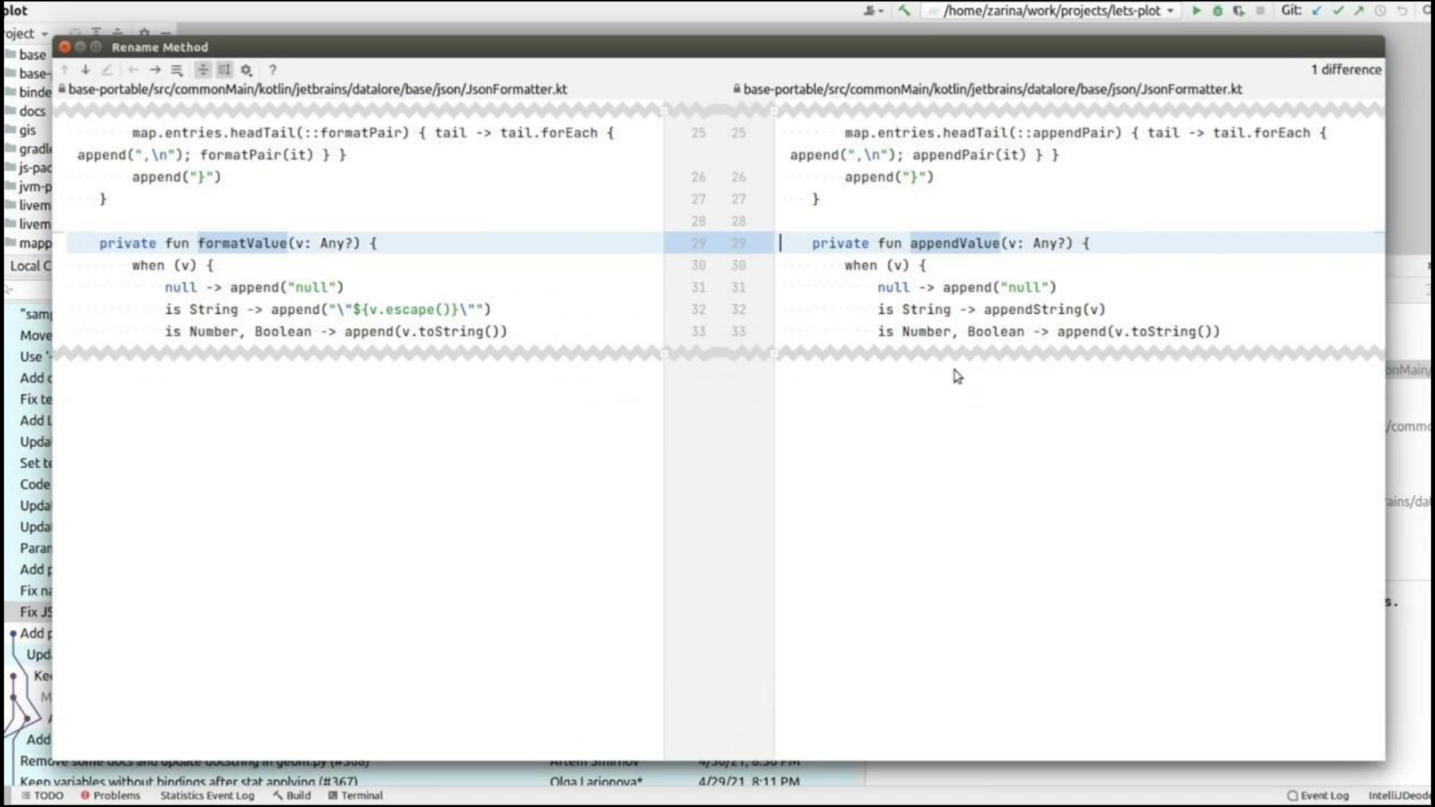
# Mining Refactorings from VCS

- Several tools exist
  - RefactoringMiner, RefDiff, Ref-Finder, …
- Perfect for empirical studies
- Could we benefit from this data within an IDE?
  - merging changes
  - data-driven code migrations
  - code reviews
  - exploring the project history
  - ...

# RefactorInsight

- Uses RefactoringMiner to detect refactorings in Java code
- Supported use cases
  - shows the list of detected refactorings in each commit or pull-request
  - shows the history of refactorings for methods and classes

# Showing the List of Detected Refactorings



/home/zarina/work/projects/lets-plot

base-portable/src/commonMain/kotlin/jetbrains/datalore/base/json/JsonFormatter.kt

```
        map.entries.headTail(::formatPair) { tail -> tail.forEach {
append(",\n"); formatPair(it) } }
        append("}")
    }

    private fun formatValue(v: Any?) {
        when (v) {
            null -> append("null")
            is String -> append("\"${v.escape()}\"")
            is Number, Boolean -> append(v.toString())
```

base-portable/src/commonMain/kotlin/jetbrains/datalore/base/json/JsonFormatter.kt

```
        map.entries.headTail(::appendPair) { tail -> tail.forEach {
append(",\n"); appendPair(it) } }
        append("}")
    }

    private fun appendValue(v: Any?) {
        when (v) {
            null -> append("null")
            is String -> appendString(v)
            is Number, Boolean -> append(v.toString())
```

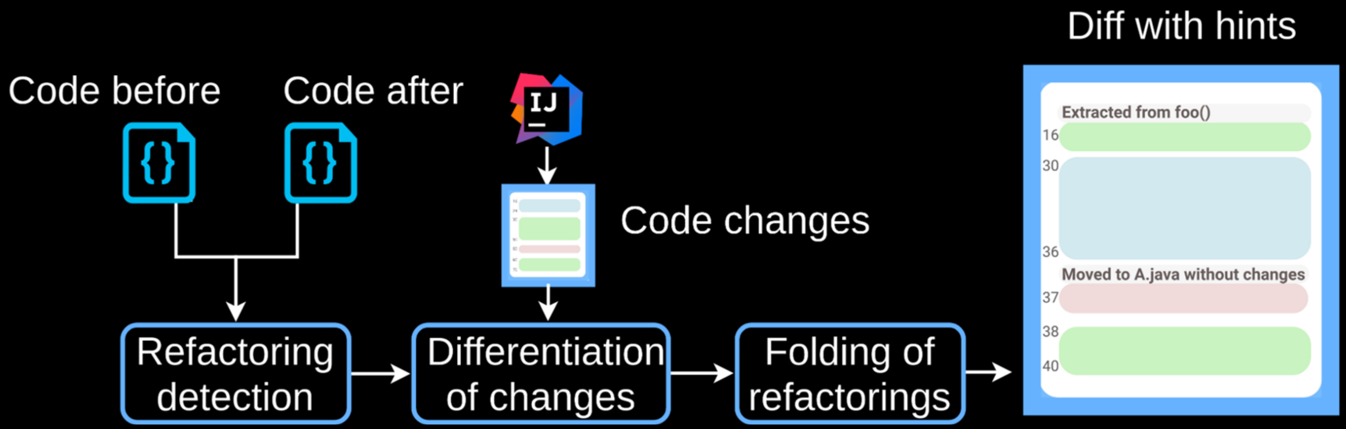
Remove some docs and update docstring in geom.py (#368) Artem Smirnov 4/30/21, 8:30 PM

Keep variables without bindings after stat applying (#367) Olga Larionova* 4/29/21 8:11 PM

# Showing the History of Refactorings for Methods and Classes

# Feedback from the IntelliJ VCS Team

- Add Kotlin support
  - developed the kotlinRMiner library
- Make the diff window aware of refactoring



Diff with hints

Code before

Code after

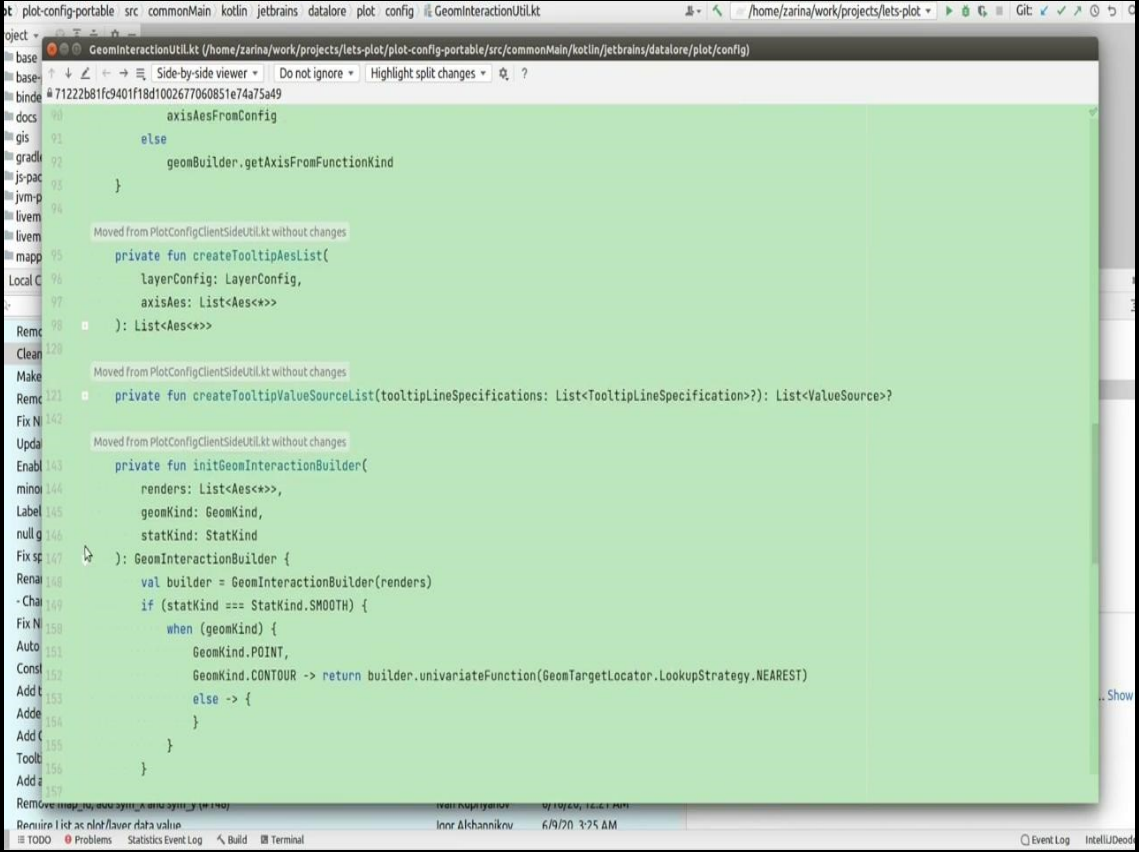
Code changes

Refactoring detection

Differentiation of changes

Folding of refactorings

# Refactoring-aware Diff Window

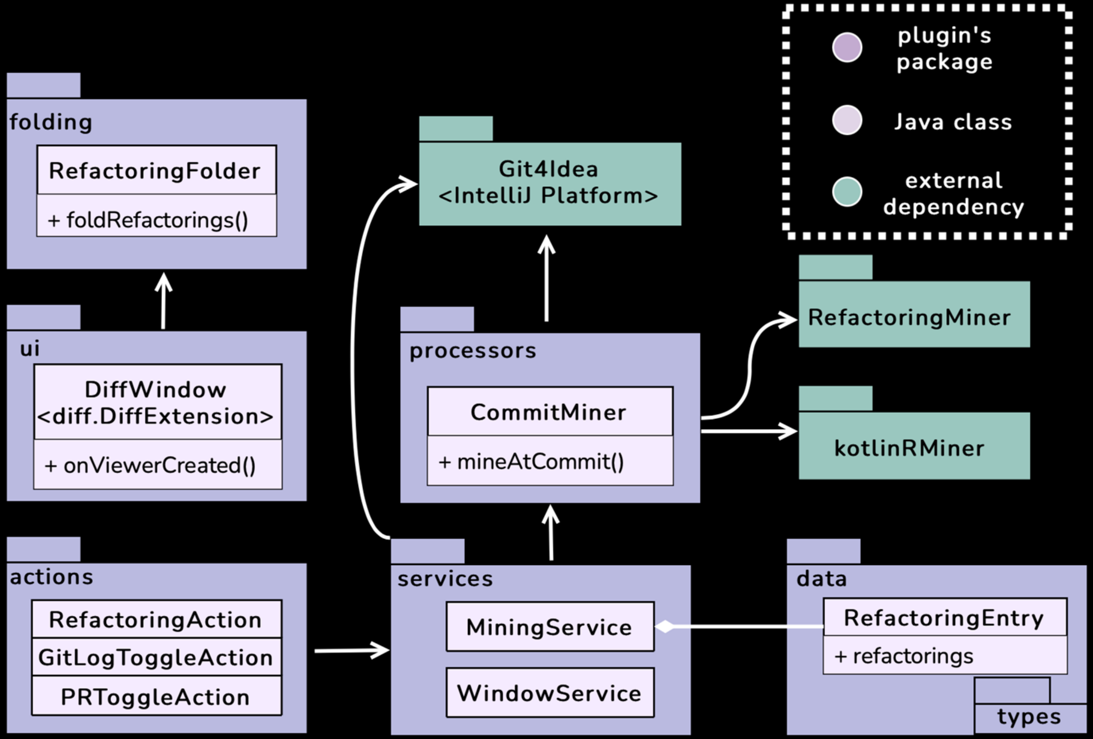


plot-config-portable › src › commonMain › kotlin › jetbrains › datalore › plot › config › GeomInteractionUtil.kt
/home/zarina/work/projects/lets-plot
Git:
GeomInteractionUtil.kt (/home/zarina/work/projects/lets-plot/plot-config-portable/src/commonMain/kotlin/jetbrains/datalore/plot/config)
Side-by-side viewer
Do not ignore
Highlight split changes
TODO
Problems
Statistics Event Log
Build
Terminal
Event Log

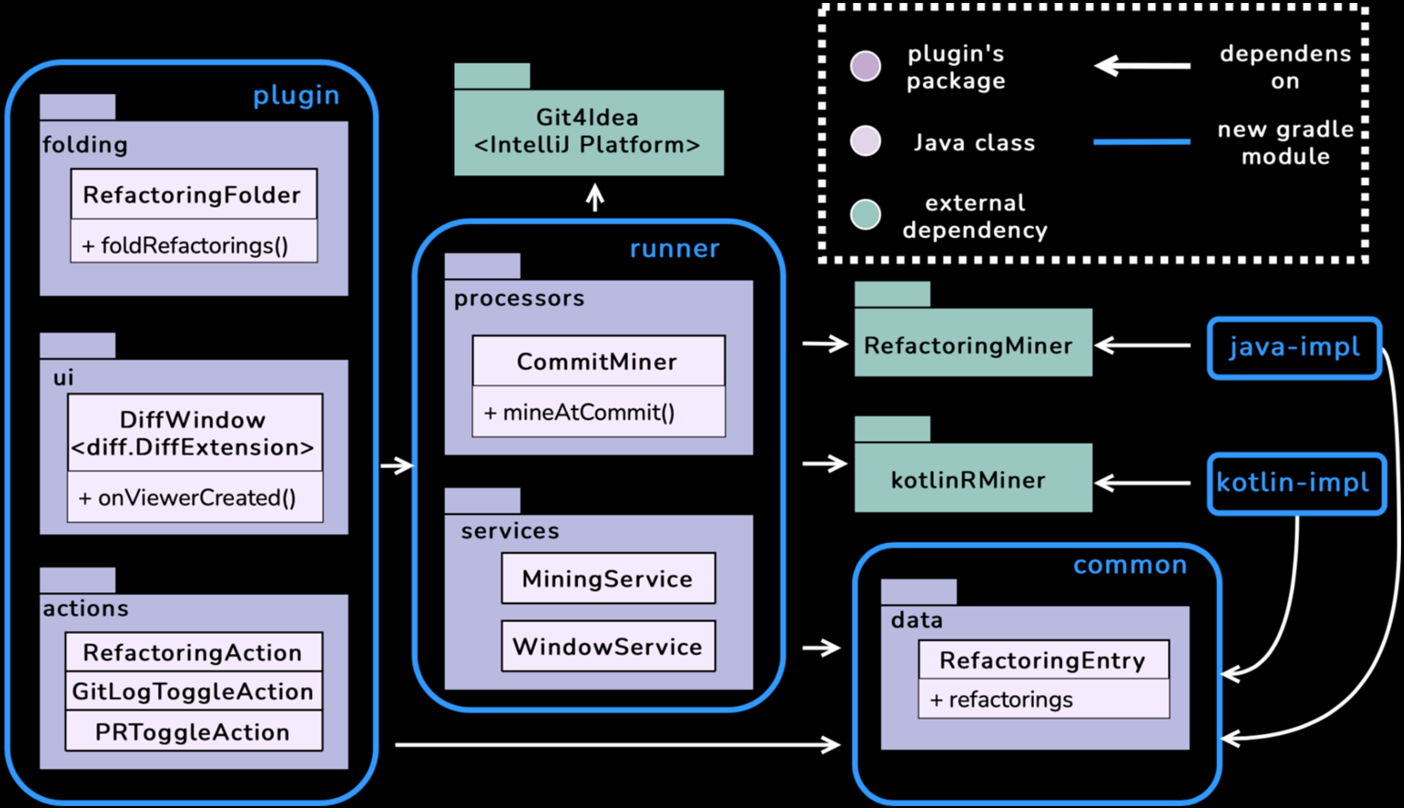
71222b81fc9401f18d1002677060851e74a75a49

```
            axisAesFromConfig
        else
            geomBuilder.getAxisFromFunctionKind
    }

Moved from PlotConfigClientSideUtil.kt without changes
    private fun createTooltipAesList(
        layerConfig: LayerConfig,
        axisAes: List<Aes<*>>
    ): List<Aes<*>>

Moved from PlotConfigClientSideUtil.kt without changes
    private fun createTooltipValueSourceList(tooltipLineSpecifications: List<TooltipLineSpecification>?): List<ValueSource>?

Moved from PlotConfigClientSideUtil.kt without changes
    private fun initGeomInteractionBuilder(
        renders: List<Aes<*>>,
        geomKind: GeomKind,
        statKind: StatKind
    ): GeomInteractionBuilder {
        val builder = GeomInteractionBuilder(renders)
        if (statKind === StatKind.SMOOTH) {
            when (geomKind) {
                GeomKind.POINT,
                GeomKind.CONTOUR -> return builder.univariateFunction(GeomTargetLocator.LookupStrategy.NEAREST)
                else -> {
                }
            }
        }
```

# RefactorInsight: Initial Architecture



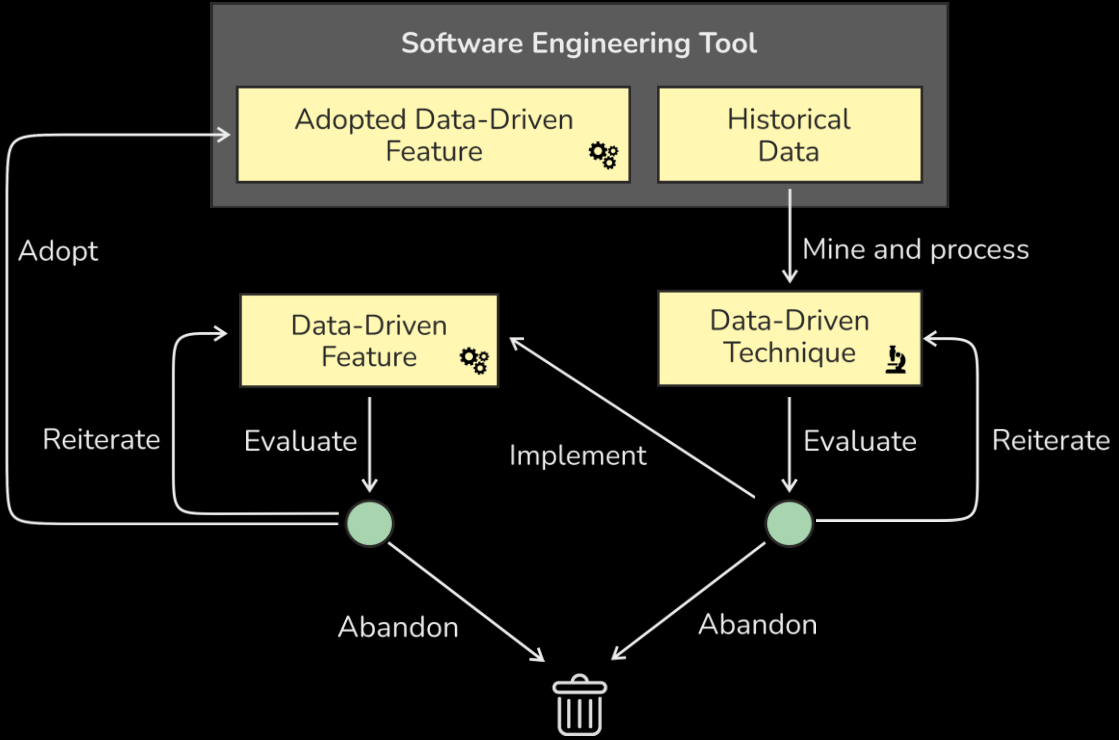
# RefactorInsight: Reworked Architecture



# Takeaways

- Production-ready research tools are rare, but they do exist
- Integrate new things into common developers workflow
  - UX should be reconsidered though
- New ideas and use cases should be explored
  - extract refactoring changes into a separate commit
  - VCS information could be helpful for refactoring recommendation as well

# Industry-Academia Collaboration



# Acknowledgements

- Nikolaos Tsantalis et al.
- Zarina Kurbatova from the ML4SE research lab
- Vladimir Kovalenko from the ICTL research lab
- Andrey Sokolov and Svetlana Zemlyanskaya from the Data Analytics team
- The whole IntelliJ VCS team
- Our wonderful interns

# Thank you!

@timofeybryksin

timofey.bryksin@jetbrains.com

https://jzuken.github.io

ML4SE Research Lab:
https://research.jetbrains.org/groups/ml_methods/

# Questions for Discussion

- Why open source is not the must in academia?
  - what should we do to make the research more reproducible?
- Is engineering less prestigious than research?
  - comparing research tracks vs industry/tool tracks
- How do you discover new ideas?