PHP static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your PHP code

Code Smell

Classes named like "Exception" should extend "Exception" or a subclass
Code Smell

Two branches in a conditional structure should not have exactly the same implementation
Code Smell

Unused assignments should be removed
Code Smell

Method arguments with default values should be last
Code Smell

A reason should be provided when skipping a test
Code Smell

"__construct" functions should not make PHP 4-style calls to parent constructors
Code Smell

PHP 4 constructor declarations should not be used
Code Smell

Deprecated predefined variables should not be used
Code Smell

"switch" statements should not have too many "case" clauses
Code Smell

Classes should not have too many methods
Code Smell

Functions should not have too many lines of code
Code Smell

Unicode Grapheme Clusters should be avoided inside regex character classes

Analyze your code

Bug | Major | regex

When placing Unicode Grapheme Clusters (characters which require to be encoded in multiple Code Points) inside a character class of a regular expression, this will likely lead to unintended behavior.

For instance, the grapheme cluster c̈ requires two code points: one for `'c'`, followed by one for the *umlaut* modifier `'\u{0308}'`. If placed within a character class, such as `[c̈]`, the regex will consider the character class being the enumeration `[c\u{0308}]` instead. It will, therefore, match every `'c'` and every *umlaut* that isn't expressed as a single codepoint, which is extremely unlikely to be the intended behavior.

This rule raises an issue every time Unicode Grapheme Clusters are used within a character class of a regular expression.

Noncompliant Code Example

```
preg_replace("/[c̈d̈]/", "X", "cc̈d̈d"); // Noncompliant, print
```

Compliant Solution

```
preg_replace("/c̈|d̈/", "X", "cc̈d̈d"); // print "cXXd"
```

Available In:

sonarlint | sonarcloud | sonarqube

"for" loop stop conditions should be invariant

Code Smell

Sections of code should not be commented out

Code Smell

Unused function parameters should be removed

Code Smell

Unused "private" methods should be removed

Code Smell