

Adding Static Analysis to Legacy Projects



Dave Liddament

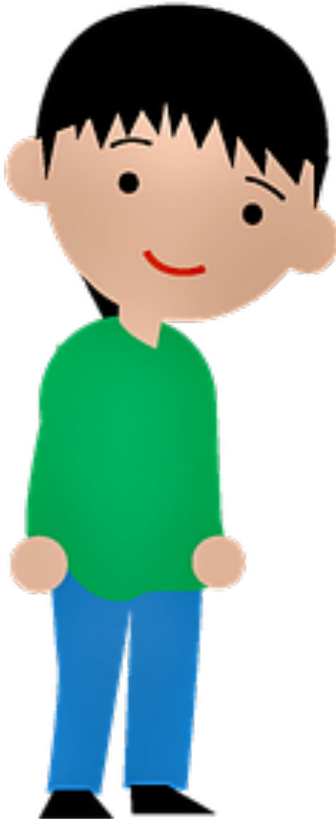
March 8th, 2023

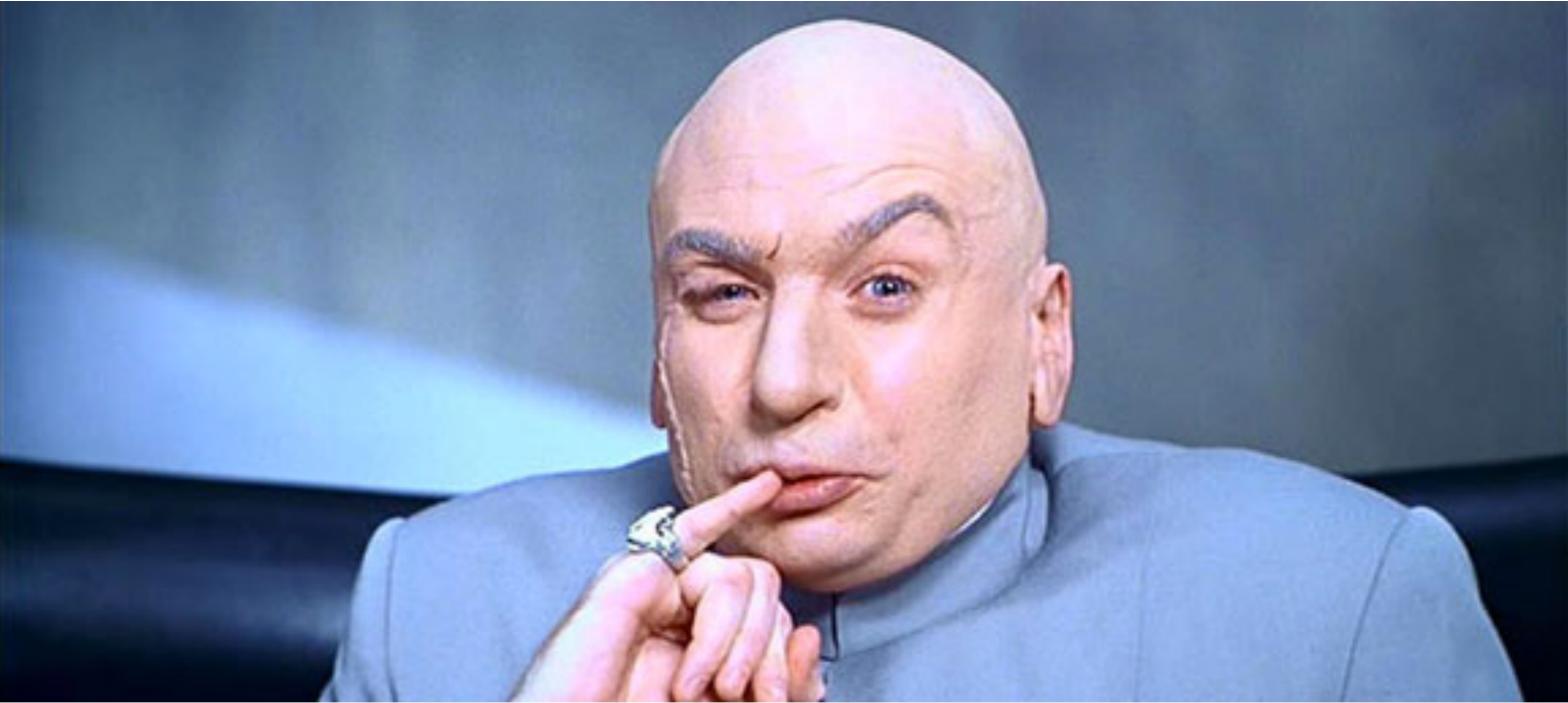
@daveliddament



I think we should try out this great new static analysis tool?

Great idea. Let's do it?



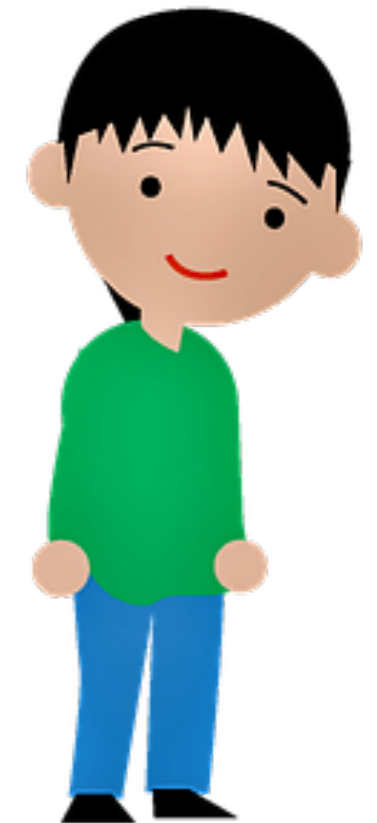


I've found 1 million issues!



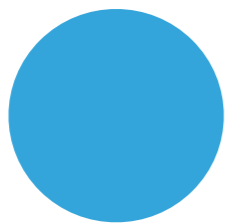
Hmm....
Well maybe next project?

Actually, there might be a tool that
can help us...



A close-up photograph of a sandy beach. A white line, possibly chalk or paint, is drawn across the sand, starting from the top left and curving towards the bottom right. The sand is a mix of light and dark brown tones, with some small dark spots and debris. The word "Baseline" is written in a large, bold, black sans-serif font across the upper right portion of the image.

Baseline



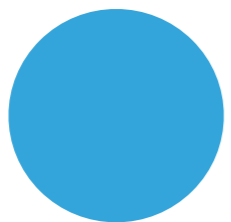
Problem

Problem

Problem

Problem

Problem



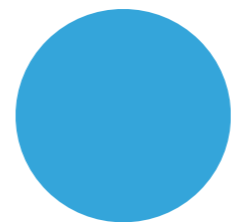
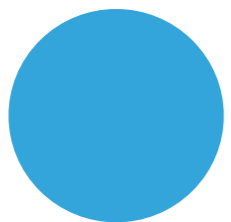
Problem

Problem

Problem

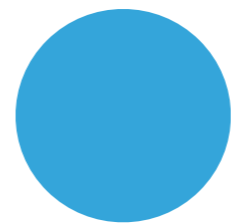
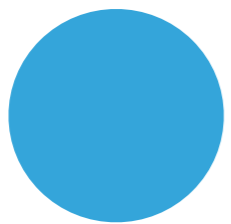
Problem

Problem

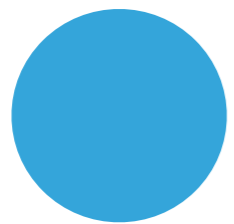


Problem
Problem
Problem
Problem
Problem
Problem

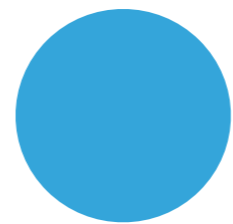
Problem
Problem
Problem
Problem
Problem
Problem
Problem



Problem
Problem
Problem
Problem
Problem
Problem



Problem
Problem
Problem
~~Problem~~
Problem
Problem
Problem



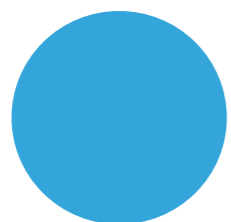
Problem

Problem

Problem

Problem

Problem



Problem

Problem

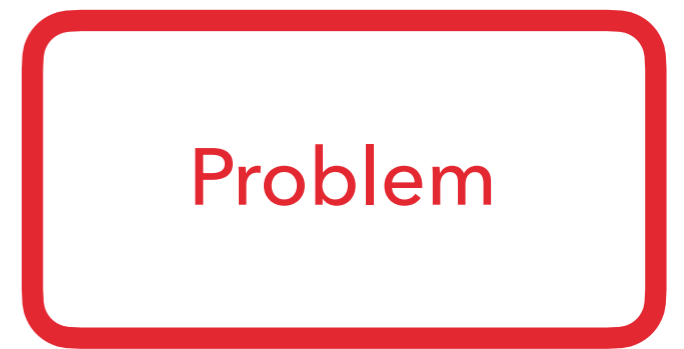
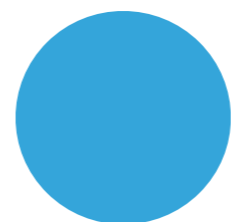
Problem

~~Problem~~

Problem


Problem


Problem




Problem


Baseline has 2,378 issues

 \leq 2,378

 $>$ 2,378

Baseline has 2,378 issues

 $\leq 2,378$

 $> 2,378$



Baseline is a file containing:

src/Entity/Person.php

- **2 * PossiblyNullReference**
- **1 * RedundantCondition**

src/Controller/Login.php

- **3 * PossiblyNullArgument**

Baseline:

`src/Entity/Person.php`

- 2 * PossiblyNullReference
- 1 * RedundantCondition

`src/Controller/Login.php`

- 3 * PossiblyNullArgument

Baseline:

src/Entity/Person.php

- 2 * PossiblyNullReference
- 1 * RedundantCondition

src/Controller/Login.php

- 3 * PossiblyNullArgument

Latest:

src/Entity/Person.php

- 3 * PossiblyNullReference
- 1 * RedundantCondition

src/Controller/Login.php

- 1 * PossiblyNullArgument
- 1 * InvalidReturnType

Baseline:

src/Entity/Person.php

- 2 * PossiblyNullReference
- 1 * RedundantCondition

src/Controller/Login.php

- 3 * PossiblyNullArgument

Latest:



src/Entity/Person.php

- 3 * PossiblyNullReference
- 1 * RedundantCondition

src/Controller/Login.php

- 1 * PossiblyNullArgument
- 1 * InvalidReturnType

Baseline:

`src/Entity/Person.php`

- 2 * PossiblyNullReference
- 1 * RedundantCondition

`src/Controller/Login.php`

- 3 * PossiblyNullArgument

Baseline:

src/Entity/Person.php

- 2 * PossiblyNullReference
- 1 * RedundantCondition

src/Controller/Login.php

- 3 * PossiblyNullArgument

Latest:

src/Entity/Person.php

- 2 * PossiblyNullReference
- 0 * RedundantCondition

src/Controller/Login.php

- 2 * PossiblyNullArgument

Baseline:

src/Entity/Person.php

- 2 * PossiblyNullReference
- 1 * RedundantCondition

src/Controller/Login.php

- 3 * PossiblyNullArgument

Latest:



src/Entity/Person.php

- 2 * PossiblyNullReference
- 0 * RedundantCondition

src/Controller/Login.php

- 2 * PossiblyNullArgument

Baseline contains:

src/Entity/Person.php

- **2 * PossiblyNullReference**
- **1 * RedundantCondition**

src/Controller/Login.php

- **3 * PossiblyNullArgument**

Baseline contains:

src/Entity/Person.php

- **2 * PossiblyNullReference**
- **1 * RedundantCondition**

src/Controller/Login.php

- **3 * PossiblyNullArgument**



Baseline contains:

src/Entity/Person.php

- **2 * PossiblyNullReference**
- **1 * RedundantCondition**

src/Controller/Login.php

- **3 * PossiblyNullArgument**



Baseline: tracking type and file

 Simple

 Can cope file renames/moves

 Fix one and introduce one

 Maintain the baseline

Static Analysis Results Baseline

- ▶ Tracks **issue type** to a **file** and **line number**
- ▶ Standalone tool can be used with any static analysis tool
- ▶ Uses git
- ▶ <https://github.com/DaveLiddament/sarb>

Static Analysis Results Baseline

- ▶ Tracks **issue type** to a **file** and **line number**
- ▶ Standalone tool can be used with any static analysis tool
- ▶ Uses git
- ▶ <https://github.com/DaveLiddament/sarb>



Git SHA: 06b982c6b3d15ef1eae827038d9d2bcb0ae71329

Type	File	Line number
InvalidNullableReturnType	src/Entity/Person.php	93
PossiblyNullReference	src/Entity/Shop.php	57
InvalidScalarArgument	src/Purchase/Begin.php	126

```
class Person
{
... Some code ...

public function foo()
{
... some code ...
    return $bar
}
```

```
class Person Employee
```

```
{
```

```
... Some code ...
```

```
public function foo()
```

```
{
```

```
... some code ...
```

```
    return $bar
```

```
}
```

```
class Person Employee
```

```
{
```

```
... Some code ...
```

Remove 20 lines of code

```
public function foo()
```

```
{
```

```
... some code ...
```

```
return $bar
```

```
}
```

```
class Person Employee
```

```
{
```

```
... Some code ...
```

Remove 20 lines of code

```
public function foo()
```

```
{
```

```
... some code ...
```

```
return $bar
```

Line 73: InvalidNullableReturnType

```
}
```


▶ **Problem:** `InvalidNullableReturnType` `src/Entity/Employee.php:73`

- ▶ **Problem:** `InvalidNullableReturnType` `src/Entity/Employee.php:73`
- ▶ What is the location of `src/Entity/Employee.php:73` at the baseline?

- ▶ **Problem:** `InvalidNullableReturnType` `src/Entity/Employee.php:73`
- ▶ What is the location of `src/Entity/Employee.php:73` at the baseline?
- ▶ History Analyser says: `src/Entity/Person.php:93`

- ▶ **Problem:** `InvalidNullableReturnType` `src/Entity/Employee.php:73`
- ▶ What is the location of `src/Entity/Employee.php:73` at the baseline?
- ▶ History Analyser says: `src/Entity/Person.php:93`
- ▶ Did we have a problem `InvalidNullableReturnType` at `src/Entity/Person.php:93` in the baseline?

Git SHA: 06b982c6b3d15ef1eae827038d9d2bcb0ae71329

Type	File	Line number
InvalidNullableReturnType	src/Entity/Person.php	93
PossiblyNullReference	src/Entity/Shop.php	57
InvalidScalarArgument	src/Purchase/Begin.php	126

Git SHA: 06b982c6b3d15ef1eae827038d9d2bcb0ae71329

Type	File	Line number
InvalidNullableReturnType	src/Entity/Person.php	93
PossiblyNullReference	src/Entity/Shop.php	57
InvalidScalarArgument	src/Purchase/Begin.php	126

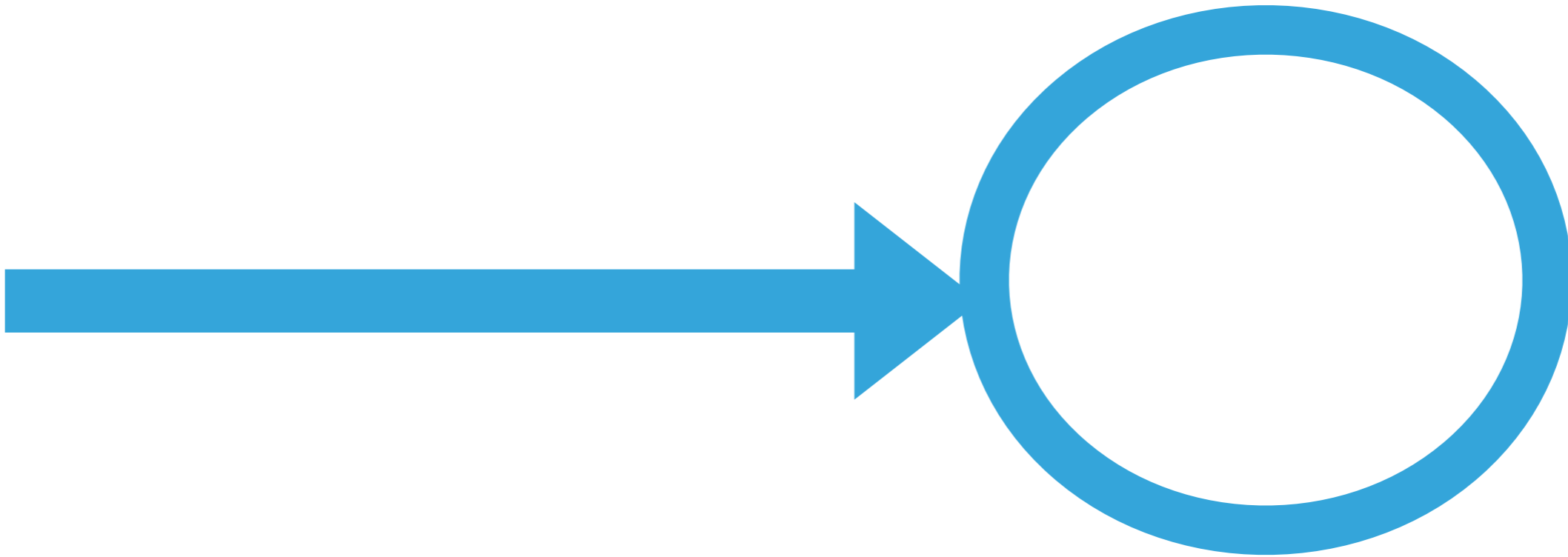
Baseline: tracking type, file and line number

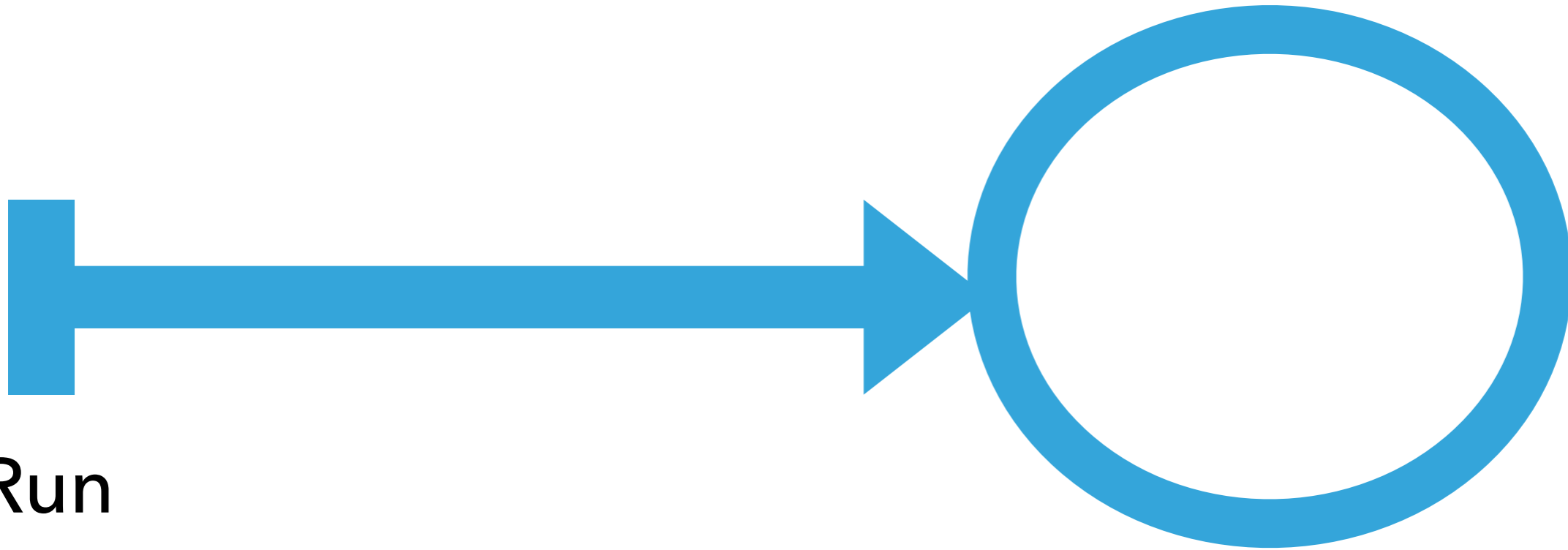
 Cope file file rename/moves

 New issues always be reported

 Complicated

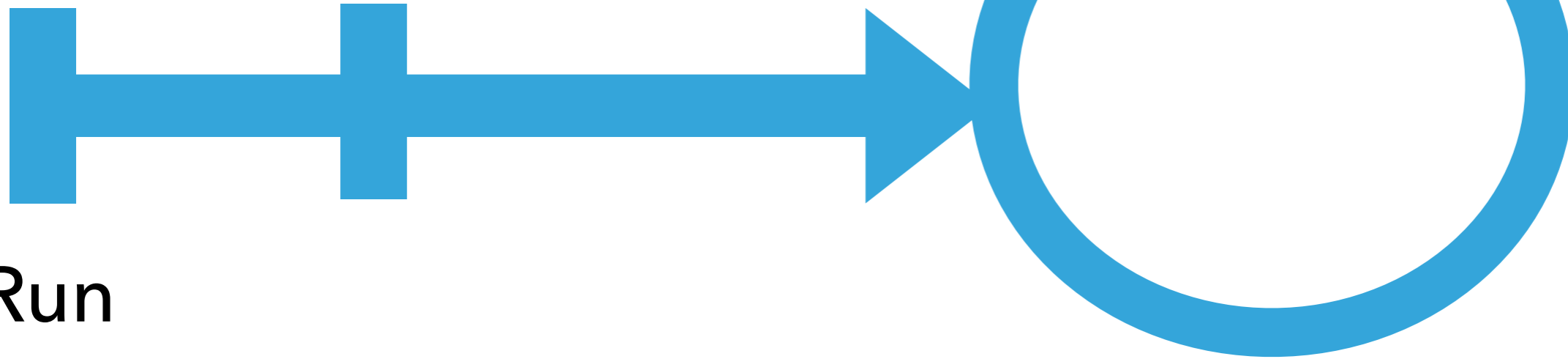
 Fix line of code with multiple issues





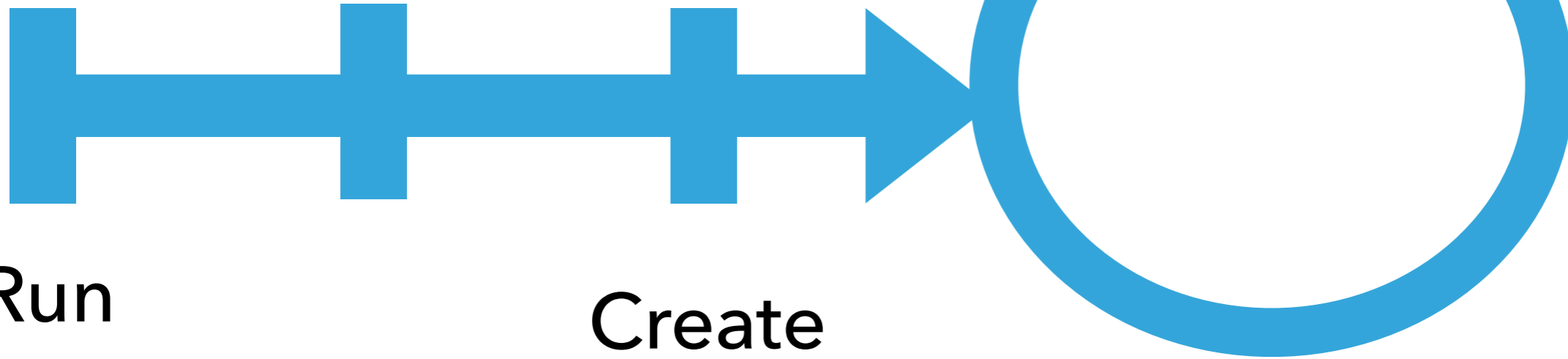
Run
static
analysis

Fix
critical
bugs



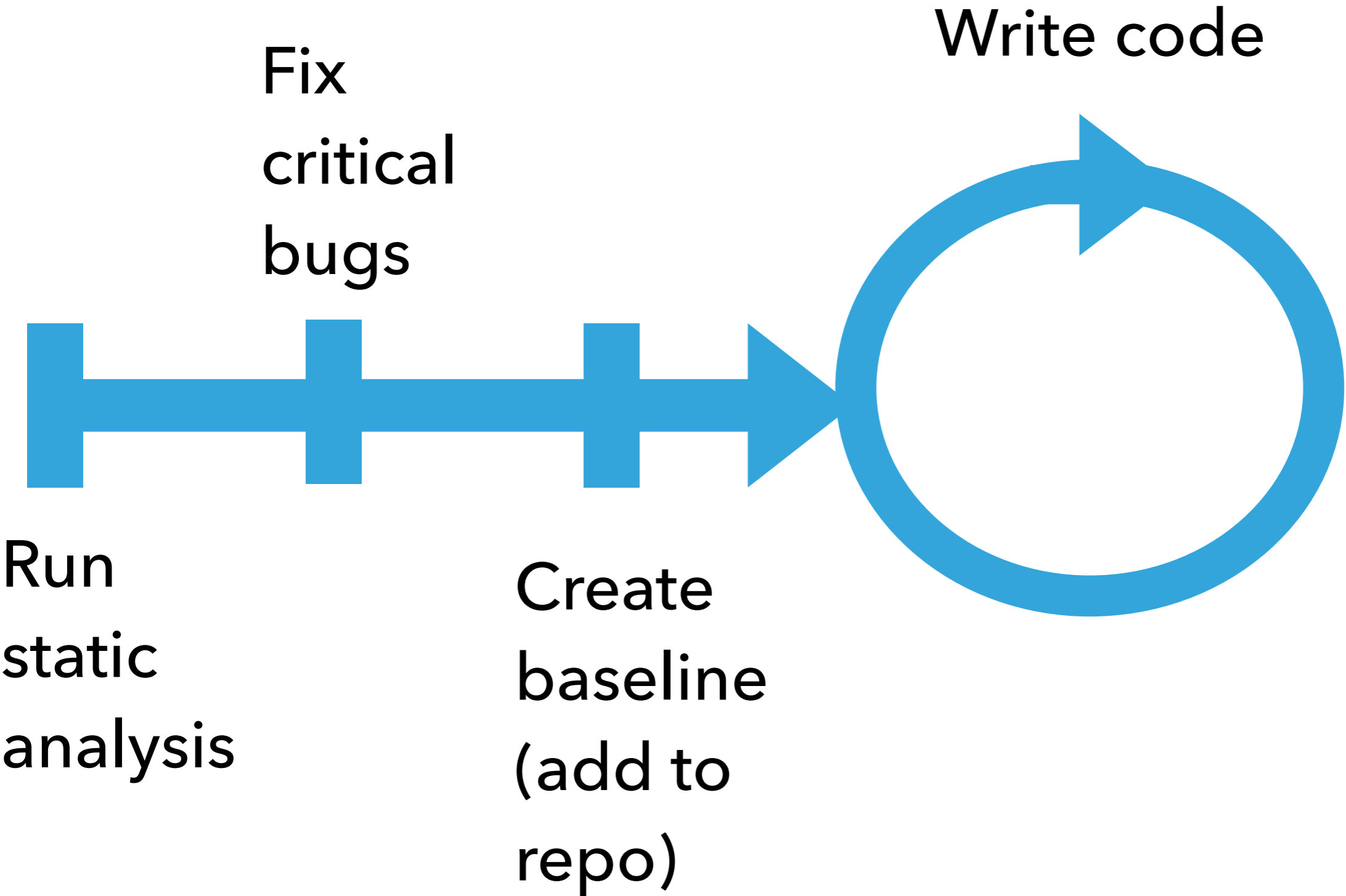
Run
static
analysis

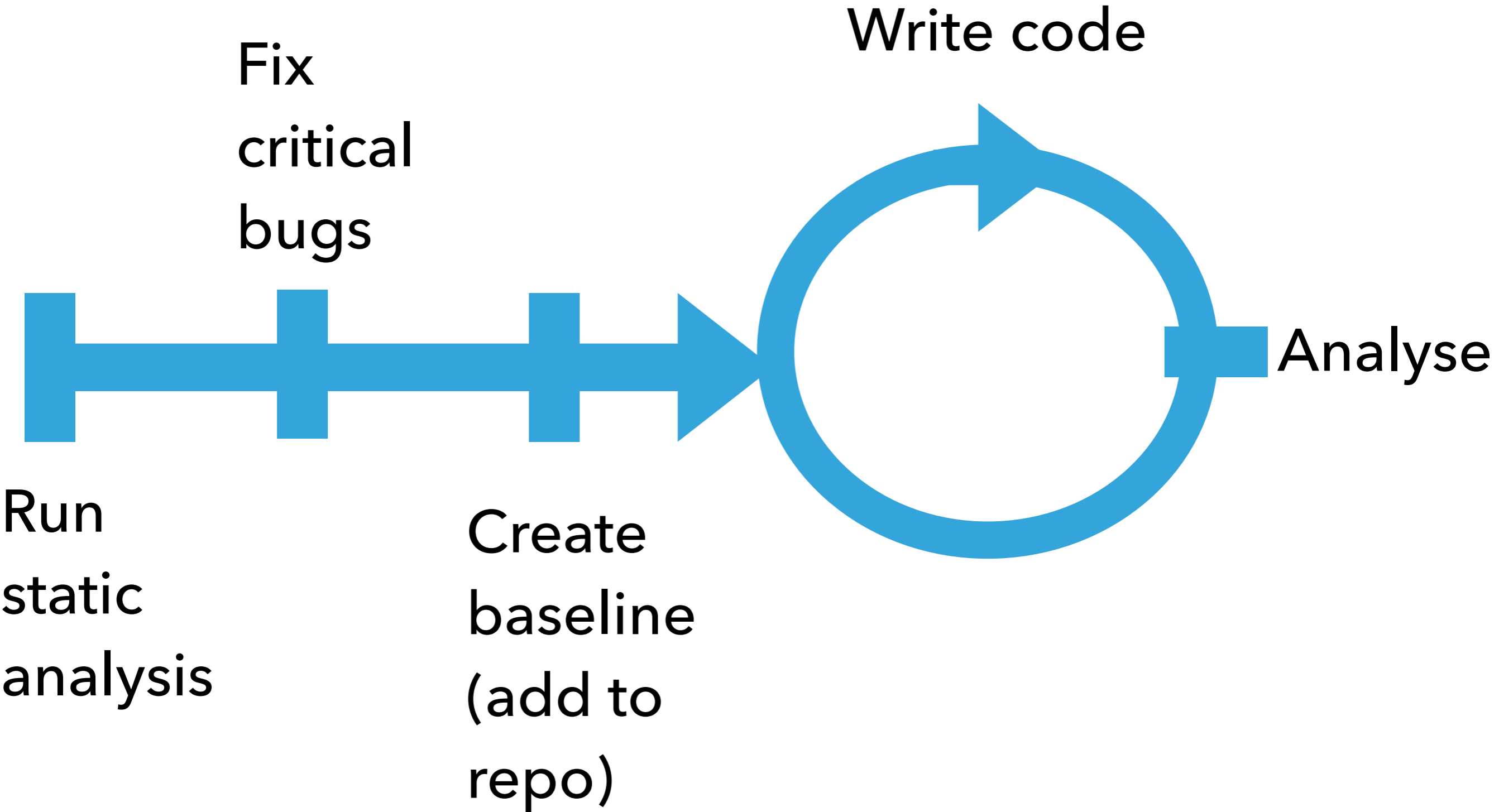
Fix
critical
bugs

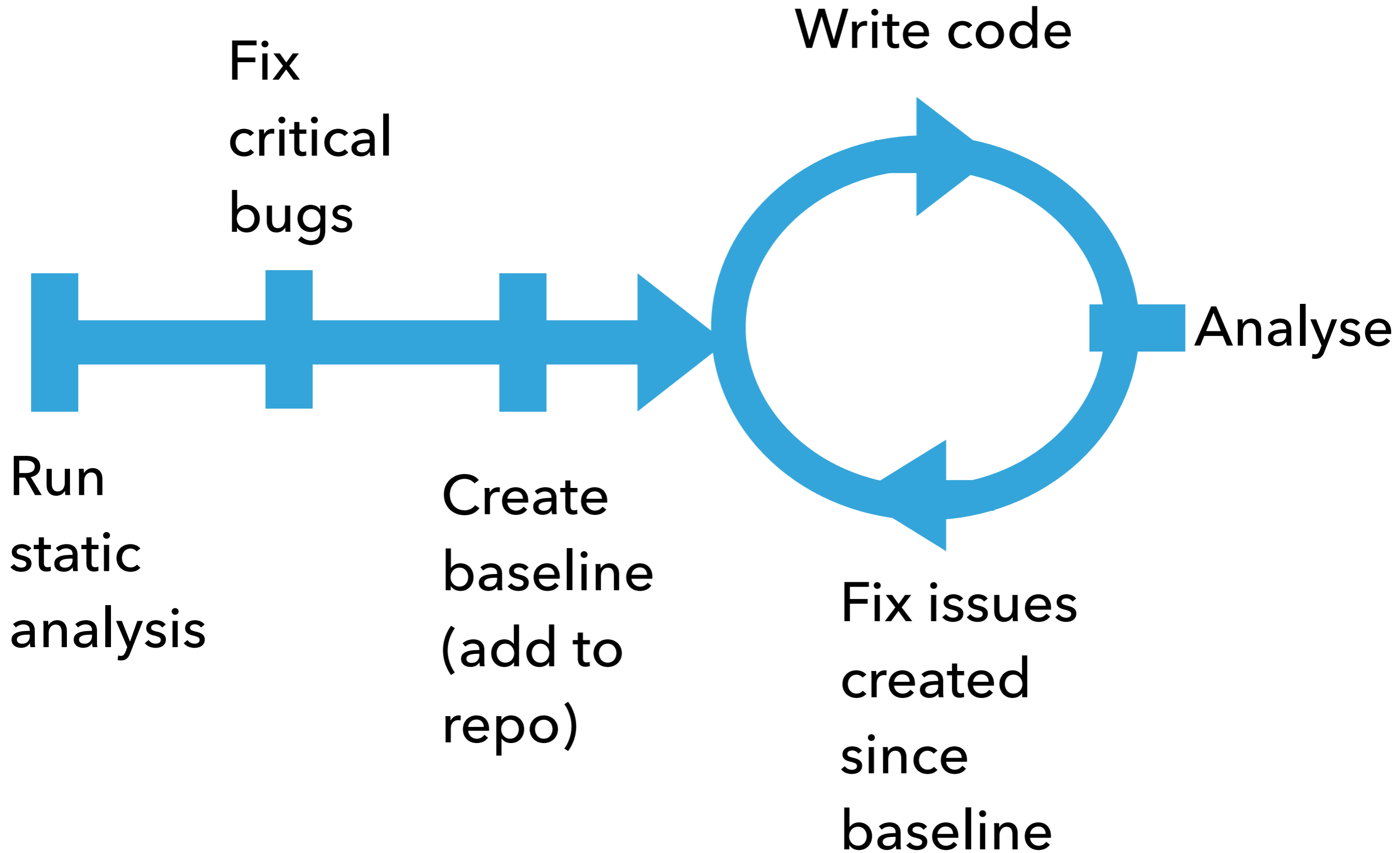


Run
static
analysis

Create
baseline
(add to
repo)









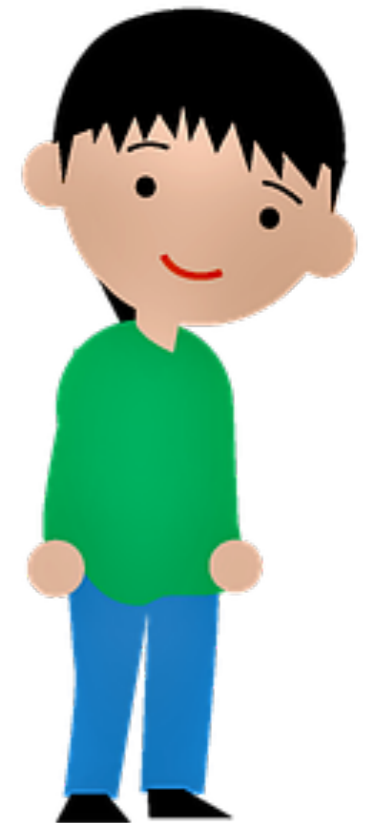
Should we ever update the baseline file once it has been created?

Yes. In these circumstances...



You'll need to update the baseline if you enable more checks...

Or upgrade the static analysis tool.



Remove entries from baseline

Original:

src/Entity/Person.php

- 2 * PossiblyNullReference
- 1 * RedundantCondition

src/Controller/Login.php

- 3 * PossiblyNullArgument

Remove entries from baseline

Original:

src/Entity/Person.php

- 2 * PossiblyNullReference
- 1 * RedundantCondition

src/Controller/Login.php

- 3 * PossiblyNullArgument

New:

src/Entity/Person.php

- 2 * PossiblyNullReference

src/Controller/Login.php

- 2 * PossiblyNullArgument

Remove entries from baseline

Original:

src/Entity/Person.php

- 2 * PossiblyNullReference

- 1 * RedundantCondition

src/Controller/Login.php

- 3 * PossiblyNullArgument

New:

src/Entity/Person.php

- 2 * PossiblyNullReference

src/Controller/Login.php

- 2 * PossiblyNullArgument

Remove entries from baseline

Original:

src/Entity/Person.php

- 2 * PossiblyNullReference

- 1 * RedundantCondition

src/Controller/Login.php

- 3 * PossiblyNullArgument

New:

src/Entity/Person.php

- 2 * PossiblyNullReference

src/Controller/Login.php

- 2 * PossiblyNullArgument

Fix ripples through the codebase

```
/** @return Person */  
function getPerson() {...}
```

Fix ripples through the codebase

```
/** @return Person */  
function getPerson() {...}
```

```
function getPerson(): ?Person {...}
```

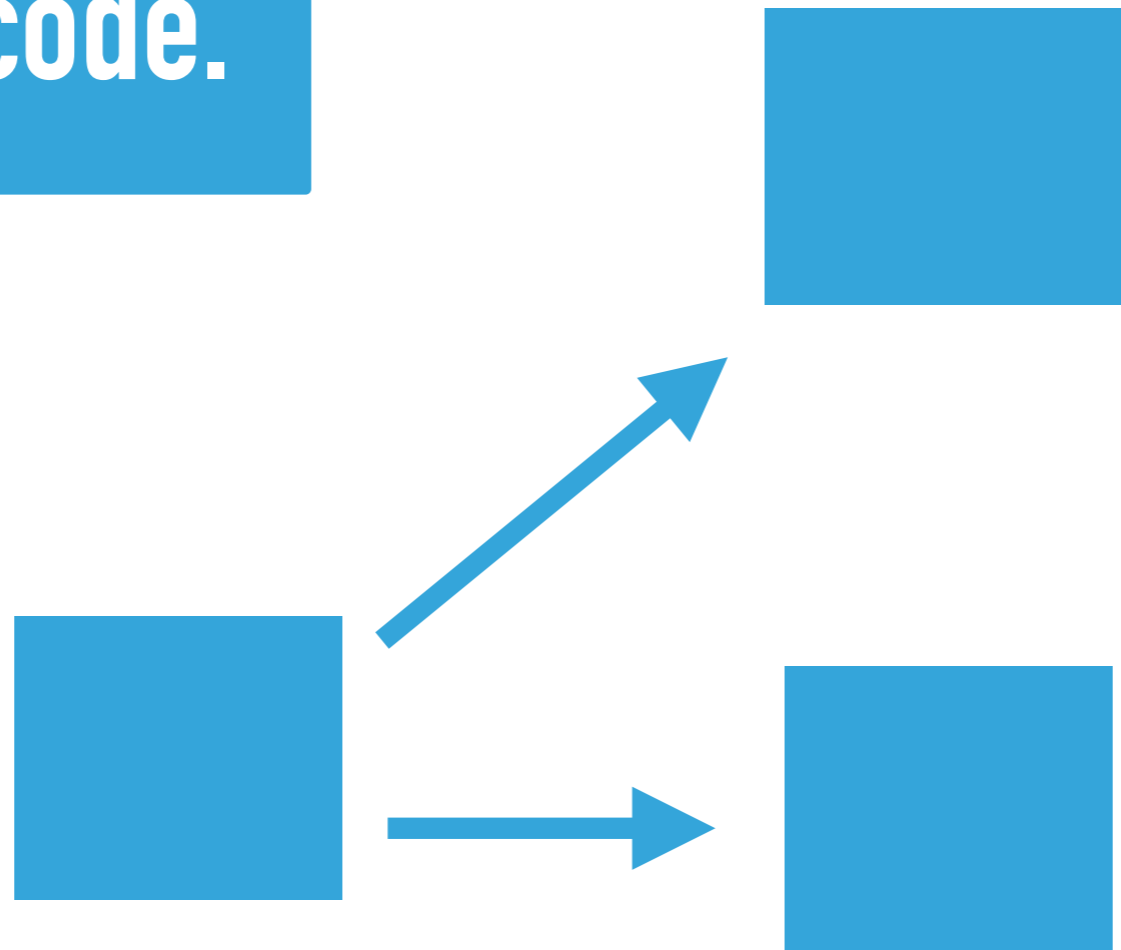
Let's change this
one line of code.



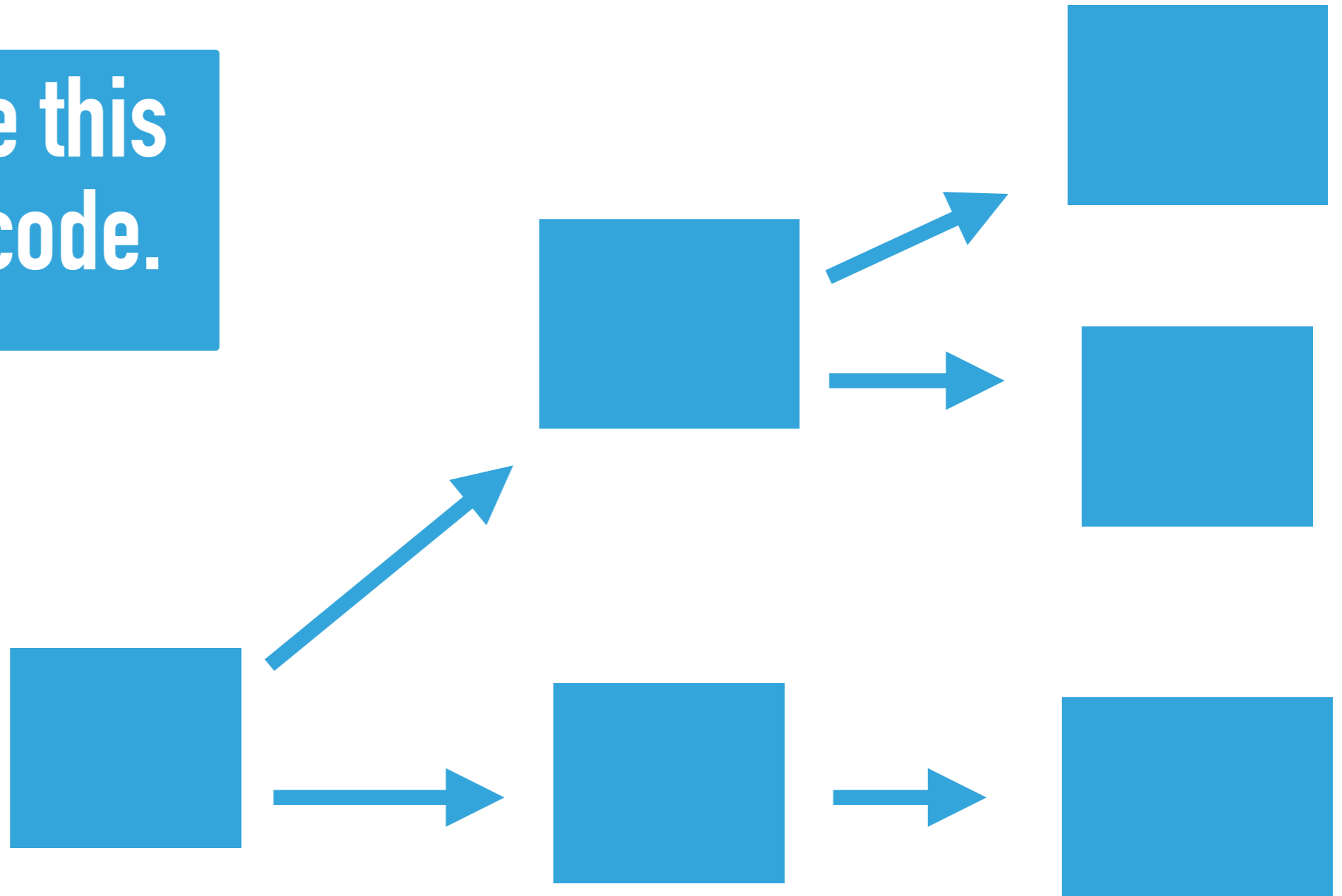
Let's change this
one line of code.



Let's change this
one line of code.

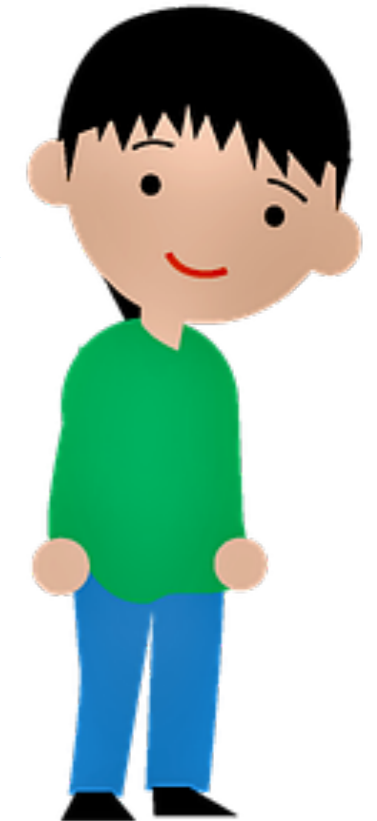


Let's change this
one line of code.





When will anyone ever address the issues in the baseline?



If you're using SARB, try it's five a day feature!



When will anyone ever address the issues in the baseline?



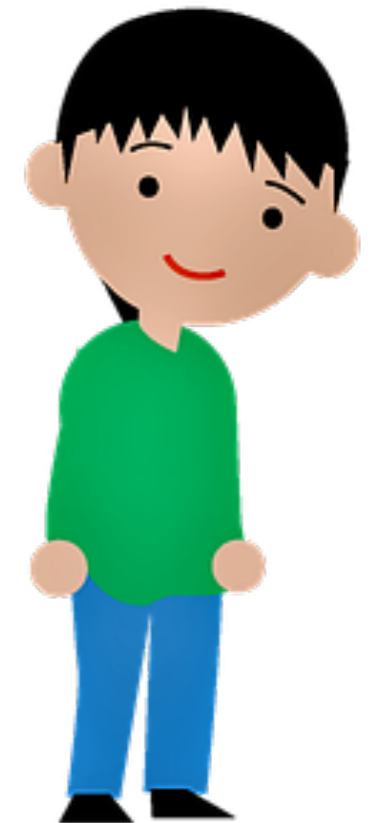
If you're using SARB, try it's five a day feature!

sarb remove phpstan.sarb **--clean-up**



I'd love to do this, but my static analysis tool doesn't have a baseline feature

Use SARB, it can be used for most tools and any language.



SARB format

```
[  
  
  {  
    "file": "/home/johnsmith/project/HomeController.php",  
    "line": 10,  
    "type": "MixedType",  
    "message": "Cannot assign $asArray to a mixed type",  
    "severity": "error"  
  },  
  
  ... repeat for each issue ...  
  
]
```

Another format

```
[
  {
    "filePath": "/home/bob/eslint/fullOfProblems.js",
    "messages": [
      {
        "ruleId": "no-unused-vars",
        "severity": 2,
        "message": "'addOne' is defined but never used.",
        "line": 1,
        "column": 10,
        "nodeType": "Identifier",
        "messageId": "unusedVar",
        "endLine": 1,
        "endColumn": 16
      },
      ... next issue in file ...
    ]
  },
  ... next file ...
]
```

```

// 1. Read JSON input from STDIN
$inputData = stream_get_contents(STDIN);
if ($inputData === false) die("Could not read input");

// 2. Convert JSON to an array
$asJson = json_decode($inputData, true);
if (!is_array($asJson)) die ("Could not parse JSON");

$issues = [];
// 3. Loop through the outer array (which has an entry for each file)
foreach($asJson as $fileWithIssues) {

    $fileName = $fileWithIssues['filePath'];
    // 4. Loop the `messages` for each file to get the file's issues.
    foreach($fileWithIssues['messages'] as $issue) {
        // 5. Add these to an array of issues.
        $issues[] = [
            'file' => $fileName,
            'line' => $issue['line'],
            'type' => $issue['ruleId'],
            'message' => $issue['message'],
        ];
    }
}

// 6. Output as JSON
echo json_encode($issues, JSON_PRETTY_PRINT);

```




Amazing. Let's start adding static analysis tools to our projects!

Great idea. Let's do it?



Thank you for
listening



Dave Liddament

@daveliddament