

Supporting Library Evolution with Smart Deprecations that Automatically Fix Client Code

Before

- 1 Library developers deprecate certain methods. Sometimes they delete methods without deprecation.



```
@Deprecated  
void method(...)
```

- 2 To update their software to the new version of a library, client developers must identify the correct replacements for deleted and deprecated methods.



500 warnings



What are the replacements for those methods?

Now

- 1 Library developers can write transformation rules to specify replacements for deprecated methods.



Do I know which methods to deprecate and what are the correct replacements?

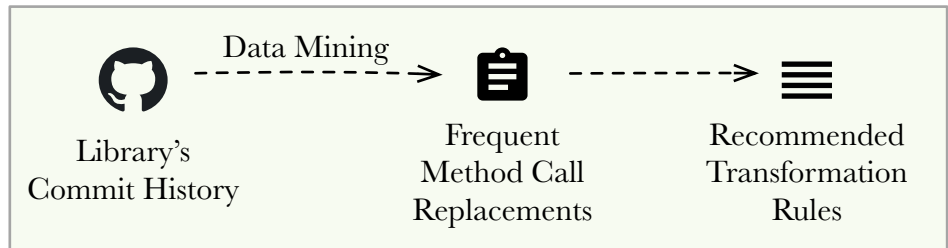
Yes

Write deprecations and transformation rules manually

```
@Deprecated  
@Transform(method -> newMethod)  
void method(...)
```

No

Ask DepMiner tool to propose deprecations and generate the rules



- 2 Client developers only have to execute their code or run the tests. When deprecated method is called, the DepRewriter tool will identify the call-site and fix client code automatically using the transformation rule. Execution will not be interrupted.



```
● test1  
● test2  
● test3
```

Invoke the deprecated method

