

## Code Reviews

### Part 2

- 
- Each team will perform the exercise (Error! Reference source not found.) bellow
- At the end, each group should make a short presentation (**Error! Reference source not found.**) about the review performed
- Updates at the end of the document

In this class the students are going to use the GitHub system, and can use the example project provided.

### Updates:

#### A. Adding Some Automation to GitHub Code Reviews

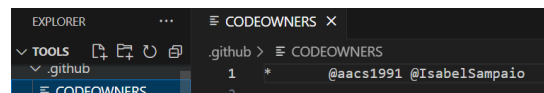
In the previous code reviews using GitHub pull requests, the author explicitly selected the reviewers. However, it is possible to automate this process. This can be done by putting reviewers' id in a specific file, named CODEOWNERS. The reviewers listed in the file must have write permissions, and the repo be public in the case of free accounts<sup>1</sup>.

Steps:

1. Create a folder named *.github* in the root of your project
2. Create a file named *CODEOWNERS* inside that folder
3. Put the reviewers' id in that file. For example insert a line like:

```
* @aacs1991 @IsabelSampaio
```

In *vscode* it would look like in the following image



4. After saving the file, put it under version control and update the repo. For example, by executing the following commands in the project directory:

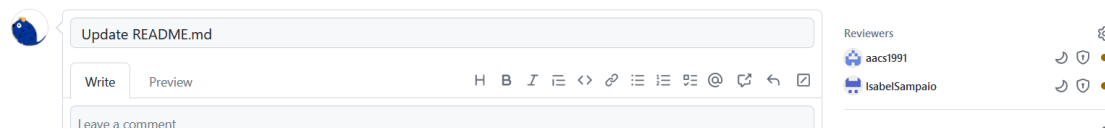
```
> git add .  
> git commit -m "CODEOWNERS file created"  
> git push origin
```

Next time a pull request is created the reviewers in the file will be automatically set as code reviewers. This is shown next for a pull request after a change of the README.md file.

---

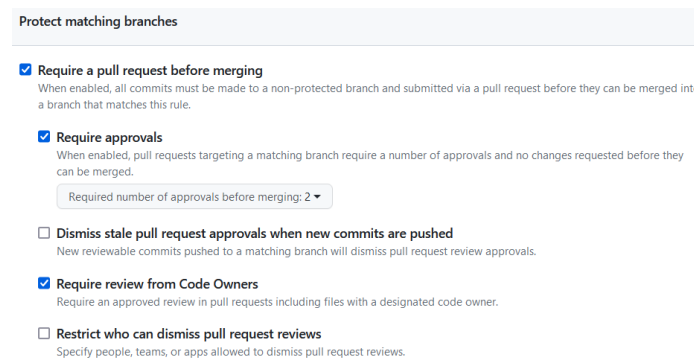
<sup>1</sup> These conditions occurred at the time of writing this document

## Switch-QA



GitHub supports more specific instructions in the file, like to indicate the kind of file changed, or use the e-mail instead of the user id – not shown here. Note that the use of the CODEOWNERS requires the repo to be public, although it can also be used by Teams created in GitHub.

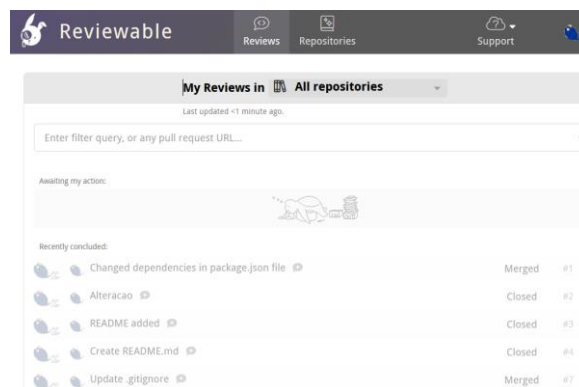
It is also possible to define rules for the branches. The user can for example force pull requests before merging. These options are available via “Settings” at the top, choose “branches” at the right and after press the button “Add branch protection rule”. See next Figure as an example.



Because “Require review from Code Owners” is enabled, merging is allowed only after the pull request is approved by people with write permission.

Beyond GitHub pull requests:

We see here code reviews using GitHub but there are other possibilities, like using special tools or other repository platforms other than github. Two examples of the latter are GitLab, which also supports the CODEOWNERS approach, and BitBucket from Atlassian (that can be used with other collaborative tools also from Atlassian). About the former, a possibility is Gerrit, a free web tool, and another is Reviewable that is integrated with github and is free for public repos. See a Reviewable screen in figure.



**B.** As an exercise, each team autonomously perform a code review using Reviewable.

**C.** Avoiding Style/Formatting Checking in the Code Review

## Switch-QA

As you know now, style/formatting is very important for code quality. A coding standard should be followed by all the members of a team or organization. Following such a standard also makes code reviews easier because no style check would be needed. Definition and use of a formatting standard can be easily done by existing tools. In this class, the Prettier tool will be used for these tasks.

Prettier is an opinionated code formatter tool. An *opiniated tool* is one that imposes a certain way of doing things, because its authors believe it is the best way to do it. The Prettier tool can be used in distinct scenarios and IDE's. Now we are going to use it with vscode.

Open vscode >> Select Extensions >> Write Prettier in the text box >> Select Prettier - Code formatter extension >> and Press install.

Now Prettier should be installed in vscode. To use it, there are two main ways, setting options in vscode settings, or externally. Anyway, the latter option seems desirable for more serious users by give you more clear control of Prettier.

Externally	Via vscode
<p>Creation of vscode configuration file:</p> <ul style="list-style-type: none"> <li>– In the project root create a folder .vscode.</li> <li>– Inside the folder create a file settings.json</li> <li>– In this file insert</li> </ul> <pre>{   "editor.defaultFormatter":     "esbenp.prettier-vscode",   "[typescript]": {     "editor.defaultFormatter":       "esbenp.prettier-vscode"   },   "editor.formatOnSave": true }</pre> <p>Now, Prettier is the default formatter and called when saving a file.</p>	<p>Open the “command Palette” with “Ctrl + Shift + P” (Cmd + Shift + P on Mac) key combination, and start writing “Format document with” and press Enter. From the drop list select Prettier.</p> <p>To use it as default code formatter go to the Settings (with Ctrl + , ) start writing “default formatter” and it will appear the option to set the default formatter. Choose Prettier as shown in the figure.</p>  <p>To format on save, in the setting, start writing “Format On Save” and enable it.</p>
<p>In any case, now create a configuration file for Prettier, named .prettierrc, and preferable, a file with what to be ignored by Prettier. Both in the project root. For example:</p>	
.prettierrc	.prettierignore
<pre>{   "trailingComma": "all",   "semi": true,   "singleQuote": false,   "printWidth": 80,   "overrides": [     {       "files": "*.json",       "options": {         "semi": false,         "tabWidth": 2       }     },     {       "files": ["*.html",         "src/*.ts"],       "options": {</pre>	<pre># Ignore artifacts: build ./dist coverage ./package-lock.json</pre>

## Switch-QA

<pre>    "tabWidth": 4   } } ] }</pre>	
--------------------------------------------------------	--

From now on, whenever you format the document (with Shift + Alt + F or by right-clicking and selecting *Format Document*) Prettier will be used. Prettier will also format the document when you save the file, so don't worry about formatting it.