



Code review changes - report

(untitled)

ID: 17

Welcome, this is the report we created for your project.

We analysed the reviews that have been merged in the last year, focusing on which changes were applied to the code under review.

We found out that you successfully merged 358 reviews!

(please, note that we restricted the scope of our tool only to the Java files in your project)

ID: 19

Code changes happening during code review in your project

How does your code change thanks to the work your team puts during the reviews? The code changes that happen during the review process are, substantially, one of the main outcomes of the code review process. We call them *review changes*. How do these review changes look like in your project?

We could classify the review changes that happened during your code review process in two main categories:

- 1. Functional changes**
- 2. Evolvability changes**

1. Functional changes impact how the program functions. A reviewer may require a functional change to the code, for instance, to check how the variables or resources are used, as well as to improve an algorithm.

Here an example from your project:

```
951 |  /**
952 |  * Optionally override the name of the upstream. If this is used, it has to
953 |  * come after any {@link #setUpstream} call.
954 |  *
955 |  * @param upstreamName
956 |  *     the name which will be used to refer to upstream in conflicts
957 |  * @return {@code this}
958 |  */
959 |  public RebaseCommand setUpstreamName(String upstreamName) {
960 |
961 |      this.upstreamCommitName = upstreamName;
962 |      return this;
963 |  }
```

Matthias Sohn can we enforce this ? Aug 15, 2012
Robin Stocker Yes, will add a check like this here: if (upstreamCommit == null) { thr... Aug 15, 2012
Robin Stocker Done Aug 15, 2012

```
951 |  /**
952 |  * Optionally override the name of the upstream. If this is used, it has to
953 |  * come after any {@link #setUpstream} call.
954 |  *
955 |  * @param upstreamName
956 |  *     the name which will be used to refer to upstream in conflicts
957 |  * @return {@code this}
958 |  */
959 |  public RebaseCommand setUpstreamName(String upstreamName) {
960 |      if (upstreamCommit == null) {
961 |          throw new IllegalStateException(
962 |              "setUpstreamName must be called after setUpstream.");
963 |      }
964 |      this.upstreamCommitName = upstreamName;
965 |      return this;
966 |  }
```

[Link to review](#)

2. Evolvability changes do not directly impact the functionality of a program, but make the software more maintainable and easier to evolve.

Examples of evolvability changes are reorganising the code, removing dead lines, and adding a comment.

ID: 20

Most of the changes that happen during the reviews in your projects are evolvability changes. For this reason, we further divided the Evolvability changes in 3 groups:

2.1 Documentation: Changes that tackle any issue in the documentation of the code. They include changes to comments and names in the program.

An example from your project:

```
438 |  /**
439 |  * Sets the fast forward mode. If set here takes precedence over the mode
440 |  * configured in git config
441 |  *
442 |  * @param fastForward
443 |  *     the fast forward mode
444 |  */
445 |  public void setFastForwardMode(boolean fastForward) {
446 |
447 |      this.fastForwardMode = fastForward;
448 |  }
```

Thomas Wolf Missing final period. Also not quite correct: if set to null, uses the def... Jun 15, 2017
Matthias Sohn Done Jun 15, 2017

```
438 |  /**
439 |  * Sets the fast forward mode. It is used if pull is configured to do a
440 |  * merge as opposed to rebase. If set here takes precedence over the fast
441 |  * forward mode configured in git config. If {@code null} uses the value of
442 |  * {@code pull.ff} configured in git config. If {@code pull.ff} is not
443 |  * configured fallback to the value of {@code merge.ff}. If also
444 |  * {@code merge.ff} is not configured --ff is the built-in default.
445 |  *
446 |  * @param fastForward
447 |  *     the fast forward mode
448 |  */
449 |  public void setFastForwardMode(boolean fastForward) {
450 |
451 |      this.fastForwardMode = fastForward;
452 |  }
```

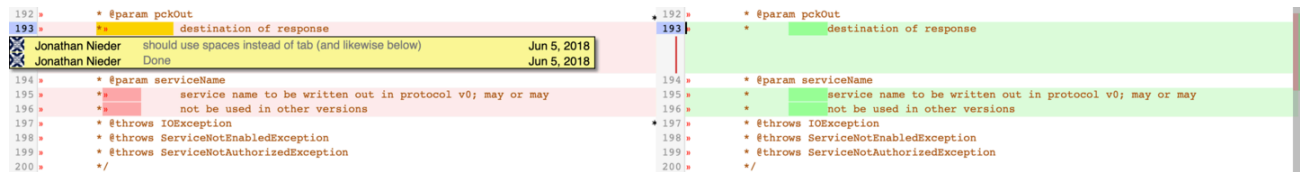
Thomas Wolf Most of this is explained below in @param. I'd just say "If non-{@code null} Jun 16, 2017
Matthias Sohn Done Jun 16, 2017

[Link to review](#)

2.2 Visual Representation: Changes that modify the layout of the code. In other

words, any formatting that does not impact the compilation result.

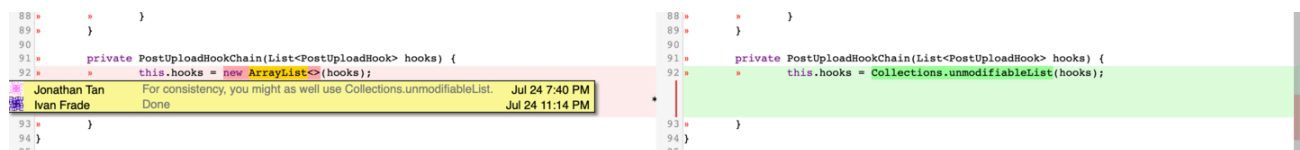
An example from your project:



```
192 * @param pckOut
193 * destination of response
Jonathan Nieder should use spaces instead of tab (and likewise below) Jun 5, 2018
Jonathan Nieder Done Jun 5, 2018
194 * @param serviceName
195 * service name to be written out in protocol v0; may or may
196 * not be used in other versions
197 * @throws IOException
198 * @throws ServiceNotEnabledException
199 * @throws ServiceNotAuthorizedException
200 */
```

[Link to review](#)

2.3 Structure: Changes to the organisation of the code itself or how a particular solution is implemented (without having an effect on the functionality of the code).



```
88 *
89 *
90
91 private PostUploadHookChain(List<PostUploadHook> hooks) {
92 * this.hooks = new ArrayList<>(hooks);
Jonathan Tan For consistency, you might as well use Collections.unmodifiableList. Jul 24 7:40 PM
Ivan Frade Done Jul 24 11:14 PM
93 *
94 }
95 }
```

[Link to review](#)

ID: 21

Visualising Review Changes Stats

This is a mock-up of our tool interface. Please take into account the space limitation; the following figure is meant to just give a general overview of the data. Our tool interface adds three new visualization on the bottom of the page.

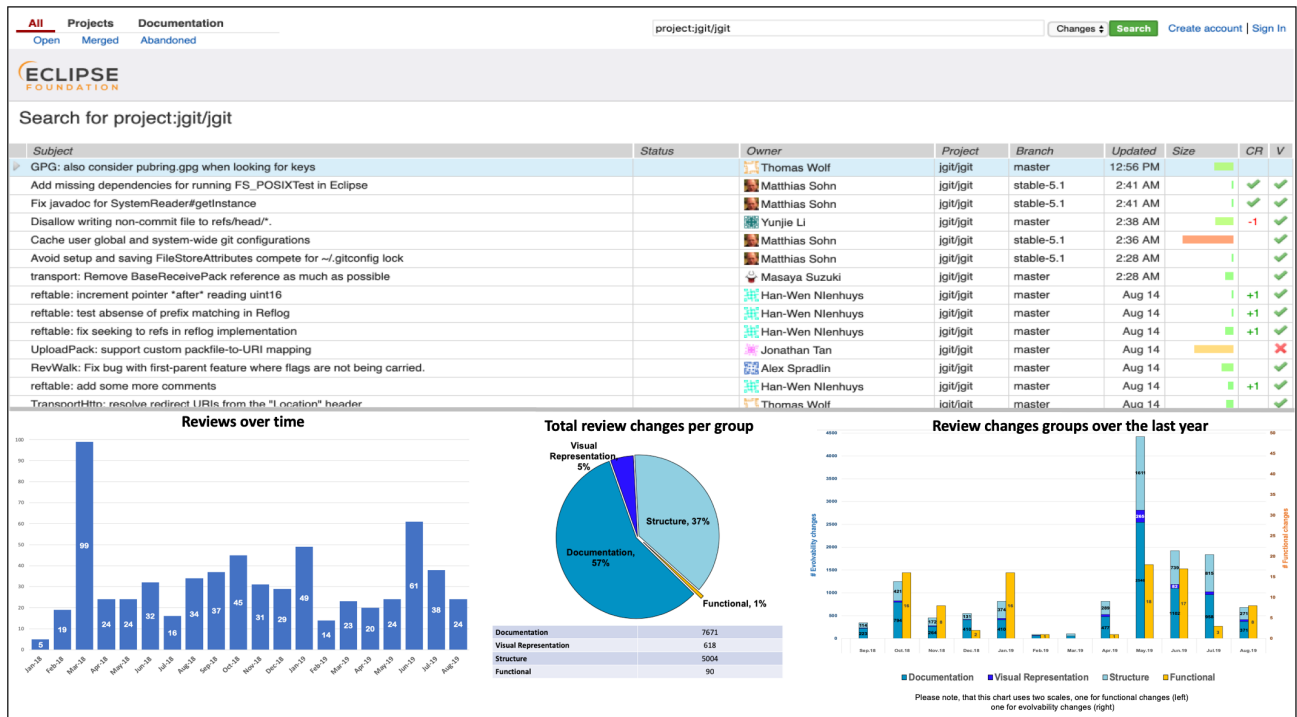
The single graphs will be shown again with more details later in the report.

The graph in the bottom left shows the number of reviews merged per month.

The graph in the middle shows the amount of review changes according to the types we have presented before: Documentation (2.1), Visual representation (2.2), Structure (2.3) and Functional (1).

The graph in the bottom right shows the distribution of the review changes over the last year.

ID: 55



ID: 22

1) After you have looked at this interface for your project, it would be great if you could let us know what you think about it:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	No Answer
The information is clear and easy to understand						
This summary is useful for my project and contains						

relevant information						
This information can be useful for a developer						
This information can be useful for a project manager						
This information reflects my expectations of the code review process of the project						

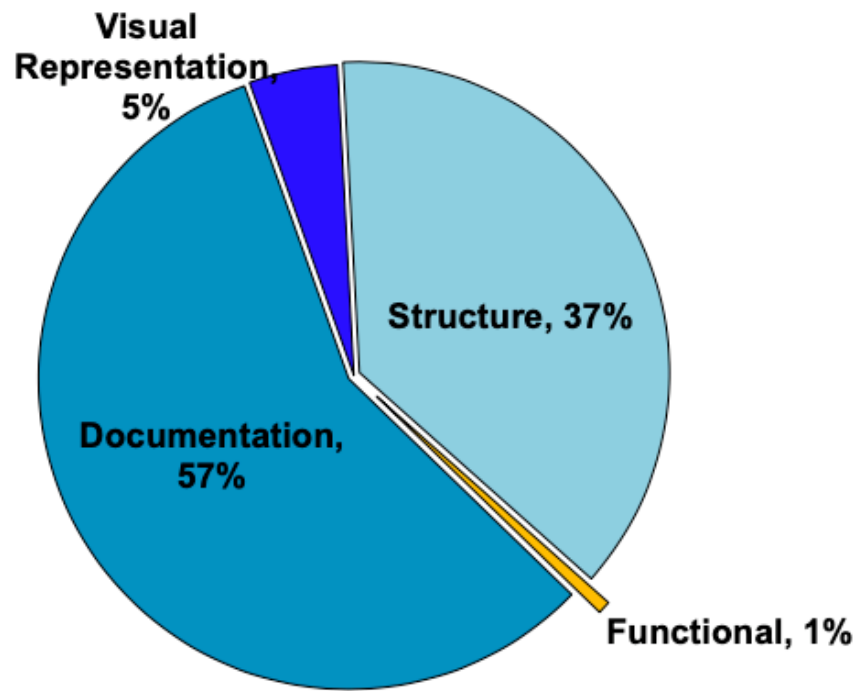
ID: 70

2) Feel free to add any further comments about your answers above or the information:

ID: 29

Details - Chart 1

This graph shows the amount of changes per group happened in your code review process from August 2019 to September 2018.



Documentation	7671
Visual Representation	618
Structure	5004
Functional	90

ID: 37

3) After you have looked at this visualization, it would be great if you could let us know what you think about it:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	No Answer
The graph is clear and easy to understand						
This summary is useful for my project and contains						

relevant information						
This information can be useful for a developer						
This information can be useful for a project manager						
This information reflects my expectations of the code review process of the project						

ID: 71

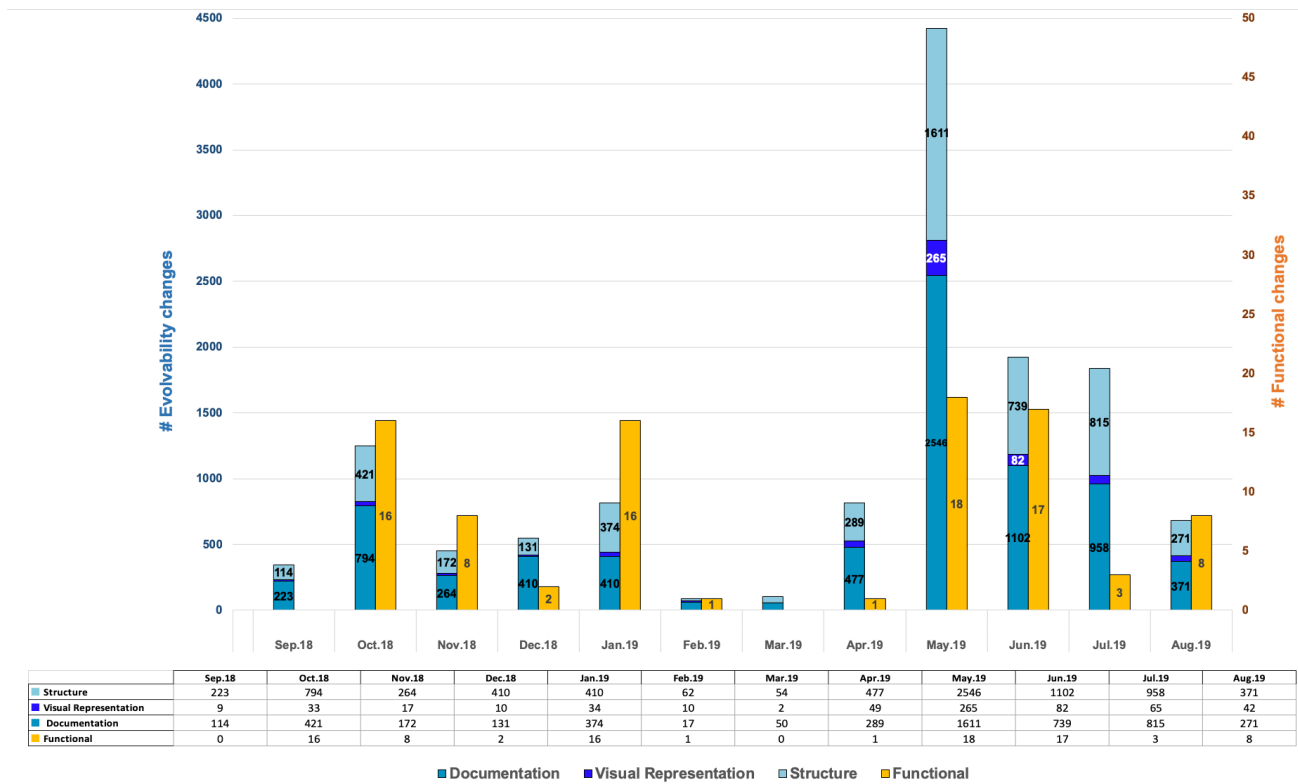
4) Feel free to add any further comments about your answers above or the visualization:

ID: 36

Details - Chart 2

This visualisation shows the amount of code review changes per group during last year.

Please take note that this chart uses two scales: one for functional changes (right), one for evolvability changes (left).



ID: 30

5) After you have looked at this visualization for your project, it would be great if you could let us know what you think about it:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	No Answer
The graph is clear and easy to understand						
This summary is useful for my project and contains relevant information						

This information can be useful for a developer						
This information can be useful for a project manager						
This information reflects my idea of the temporary distribution of the changes in our review						

ID: 72

6) Feel free to add any further comments about your answers above or the visualization:

ID: 50

Considering all the information seen so far, it would be great if you could give us some feedback by answering the next few questions:

Logic: Show/hide trigger exists.

ID: 3

7) The whole report is useful for my project and contains relevant information.

Strongly Disagree Disagree Neutral Agree Strongly Agree No Answer

Logic: Hidden unless: #7 Question "The whole report is useful for my project and contains relevant information." is one of the following answers ("Agree","Strongly Agree")

ID: 73

8) Please specify:

Logic: Show/hide trigger exists.

ID: 6

9) I have learned something about my project that I have not been aware of before.

Strongly Disagree Disagree Neutral Agree Strongly Agree No Answer

Logic: Hidden unless: #9 Question "I have learned something about my project that I have not been aware of before." is one of the following answers ("Agree","Strongly Agree")

ID: 74

10) Please specify:

ID: 7

11) The results made me curious. I plan to investigate further the code review process of my project.

Strongly Disagree Disagree Neutral Agree Strongly Agree No Answer

ID: 15

12) This report provides information that is not available in any other tool I know.

Strongly Disagree Disagree Neutral Agree Strongly Agree No Answer

ID: 16

13) Having this kind of information at ready would have a positive influence over the Code Review process of my team.

Strongly Disagree Disagree Neutral Agree Strongly Agree No Answer

ID: 57

14) Based on the information you have seen in this report, I would change something in the code review practices of my project:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	No Answer
Code review policies (e.g., mandatory review after each commit)						
Use of tools (e.g., static analysis tools like Check style or FINDBUGS)						
Reviewers assignment (e.g., by finding specific reviewers for specific defects)						

Logic: Hidden unless: ((Question "Code review policies (e.g., mandatory review after each commit)" is one of the following answers ("Agree","Strongly Agree") OR Question "Use of tools (e.g., static analysis tools like Check style or FINDBUGS)" is one of the following answers ("Agree","Strongly Agree")) OR Question "Reviewers assignment (e.g., by finding specific reviewers for specific defects)" is one of the following answers ("Agree","Strongly Agree"))

ID: 64

15) Please, give us further details on this:

ID: 66

16) I think that the information of this report might help with ...

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	No Answer
To understand how resources are spent on our project						
To avoid performing the same mistakes again						

Logic: Hidden unless: (Question "To understand how resources are spent on our project" is one of the following answers ("Agree","Strongly Agree") OR Question "To avoid performing the same mistakes again" is one of the following answers ("Agree","Strongly Agree"))

ID: 75

17) Please, give us further details on this (*optional*):

ID: 8

18) If you want, please give further feedback on our report. (Optional)

ID: 53

Background

To improve the scientific validity of your feedback, It would be great if you could answer to the following demographic questions too.

ID: 81

19) What is your involvement with the development of the project analyzed in this report?

I am a core developer

I have committing/merging rights

I am an external contributor (no merge rights)

I am an active user (e.g., I open issue reports, but do not develop any code in it)

I am an external user (I am not involved in the development at all)

Other - Write In:

ID: 11

20) What is your current job?

Professional Developer

Academic Researcher

Industrial Researcher

Project Manager

Spare-time Developer

Student

Other - Write In:

ID: 13

21) How many years have you developed software in a professional setting?

I have no experience in software development in a professional setting 1 year or less 2 years 3-5 years 6-10 years 11 years or more

ID: 14

22) How often do you currently program software?

Not at all About once a year About once a month About once a week Daily or more often

ID: 51

23) How many years have you performed code reviews?

I have never done code review 1 year or less 2 years 3-5 years 6-10 years 11 years or more

ID: 52

24) How often do you currently perform code reviews?

Not at all About once a year About once a month About once a week Daily or more often

ID: 77

In addition to analyzing your data in an anonymous form, can we also share your answers (in an anonymous form) in a publicly available dataset?*

(Selecting "yes" allows other researchers and the public to benefit from your answers and effort)

Yes

No

ID: 79

25) If you have any comments on the report or on the topic, please add them here.

ID: 1

Thank you very much for reading our report. We hope it gave you useful insights!
Also, we thank you for the time you spent giving us feedback.

You are awesome!

**If you are interested in the detailed data that we used to create this report, don't
hesitate to contact us at: *email address***
