

Block 1

Welcome!

We are a group of researchers from the University of Zurich investigating ways to improve code review.

We would really appreciate if you could help us by participating in our survey.

Code review is a part of the software development cycle to improve code quality, i.e. the detection of bugs, knowledge transfer and team awareness. However, Code review can easily become a complex task.

To support reviewers, a team of researchers has created a visualization tool for code reviews called ReviewVis.

In this survey, we would like you to evaluate this tool that they created.

The survey will last approximately 15 minutes and has a total of 10 pages. As an appreciation of your help for our research, we will donate 2 USD to a non-profit organisation on your behalf (more about this in the next page).

Thank you for your help!

Data policy consent

In the survey, we ask you some demographic data such as your level of education, gender, and your experience as a developer; none of these questions will be mandatory and we will not ask any further sensitive information. Please do not to share personal and/or sensitive information in the open questions you will find in the survey. In such a case, we will remove your answer from the final dataset.

Your answer are anonymized to the best of our capabilities, moreover the IP address from which you complete the survey is not saved. You can withdraw from the survey at any moment. At the beginning of the survey, you will be provided an unique ID (combination of an endangered species, an english emotion and a random number). This ID will be used in case you want us to remove your answer from the dataset, whether you completed the questionnaire or not.

By proceeding to the next page you agree to our data handling policy.

Donation

Donation on your behalf

To thank you for your participation in the experiment, we are going to donate 2 USD (or its equivalent) to a non-profit organization on your behalf.

The donation is done only on the behalf of participants whose experiment is valid and complete.

The donation will be done at the end of the entire study and will be announced from [this Twitter Account.](#)

Please select the organization you prefer:

- ☐ University Hospital Zurich Foundation
- ☐ Wikimedia Foundation
- ☐ Mozilla Foundation

- ☐ Médecins Sans Frontières
- ☐ Electronic Frontier Foundation
- ☐ WWF
- ☐ UNICEF
- ☐ The UN Refugee Agency
- ☐ Amnesty International

Block 5

This is the ID assigned to you: \${e://Field/ResponseID}

If you keep a copy of the following unique code, you can use it to request the removal of your responses from this survey at any time.

Default Question Block

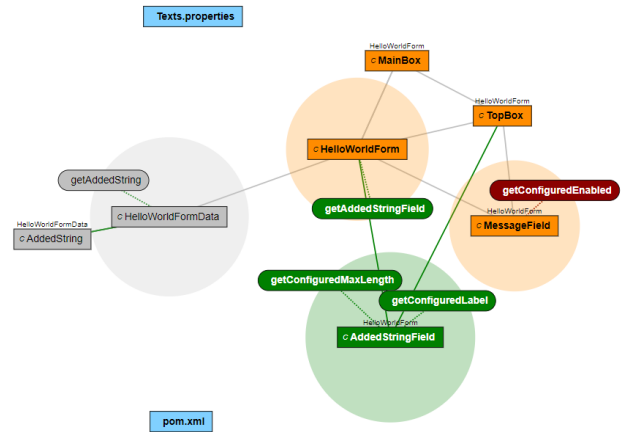
This is a short overview about the developed tool ReviewVis. We will explain to you the visualization and its features step by step.

ReviewVis works in addition to GitLab merge requests, a web based tool to perform code review on the changes. ReviewVis opens in a separate window and draws the graph according to the changes in the merge request.

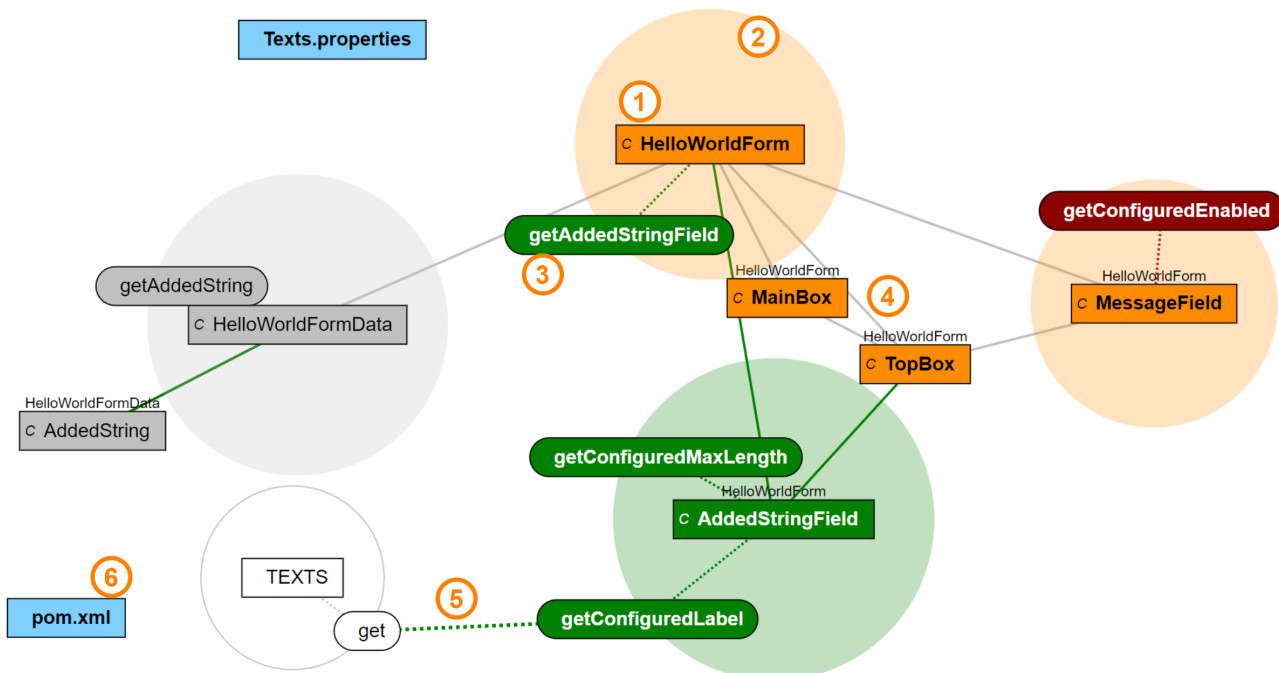
During the survey you will not be able to try the prototype of ReviewVis. Its features will be illustrated through videos. Please keep this into account while answering the questions.

The code review example is taken from the [Eclipse Scout](#) project. Here is a short preview of how ReviewVis integrates with GitLab:

The screenshot shows a Git merge request (MR) titled "Test mr" opened 2 months ago by Administrator. The interface includes tabs for Overview, Commits, Pipelines, and Changes. The "Changes" tab is active, showing a diff for the file `scout.client/src/main/java/ch/uzh/scout/client/helloworld/HelloWorldForm.java`. The diff highlights several changes, including imports, annotations, and method implementations. The right sidebar shows the file structure and a list of changes.



The graph contains nodes that represent classes, interfaces and methods. Links between nodes represent any kind of dependencies (for example, class / interface hierarchies, or method calls).



1. represents a Java class. Classes are denoted with C, Abstract classes with A and Interfaces with I
2. the circle is added to nodes that contain at least one method in the code change
3. rounded corners of a node indicate a method

4. inner classes are denoted with a label on the top with the corresponding class file name
5. represents a method call. "getConfiguredLabel" of class "AddedStringField" calls "get" from "TEXTS"
6. nodes that do not represent Java components are shown with their full name

The (change-based) color scheme for nodes and links is: added, changed, deleted, generated, unchanged, non-Java

Please answer the following questions about the information displayed with the graph of ReviewVis.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	I don't know
The graph is easy to understand	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
In the graph, the colour-coding of the nodes is clear	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Block 10

In the video below, you see the layout features of ReviewVis:

- *Zoom* in and out of the graph
- *Drag* single nodes
- Nodes and spheres do not overlap on *collision*
- Connected nodes are highlighted on *hover*

In the subtitles we will display relevant information about the feature. Please make sure they are active.

layout feature



It is important that the explanation of the feature is clear. So, please answer the following question:

Where are the nodes representing methods positioned?

- ☐ Outside the circle of a class but close to the class in which they are implemented
- ☐ Inside the circle of a class
- ☐ Randomly in the canvas

Please evaluate the following claim on ReviewVis' layout features:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	I don't know
The layout feature of ReviewVis would help me better understand a change to review	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Why do you think the layout feature would support your understanding of the code to review?

Why do you think the layout feature would **not** support your understanding of the code to review?

Please evaluate the following claim on ReviewVis' layout features:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	I don't know
The interaction offered by this feature would support me in understanding a change to review	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Block 9

In the video below, you see the *graph-to-code* features of ReviewVis:

- Clicking on a node *jumps to the line* of source code
- Not yet clicked nodes are highlighted with a thicker border to *track the progress* of the review

In the subtitles we will display relevant information about the feature. Please make sure they are active.

navigation feature



It is important that the explanation of the feature is clear. So, please answer the following question:

For which nodes in the graph does the graph-to-code functionality work?

- ☐ Only for class nodes
- ☐ Only for method nodes
- ☐ For both method and class nodes

Please evaluate the following claim on ReviewVis' graph-to-code features:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	I don't know
The graph-to-code feature of ReviewVis would help me better understand a change to review	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Why do you think the graph-to-code feature would support your understanding of the code to review?

Why do you think the graph-to-code feature would **not** support your understanding of the code to review?

Please evaluate the following claim on ReviewVis' graph-to-code features:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	I don't know
The interaction offered by this feature would support me in understanding a change to review	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Block 8

In the video below, you see the *code-to-graph* features of ReviewVis:

- Hovering over a line of code in the source code jumps to the node and highlights its neighbors

In the subtitles we will display relevant information about the feature. Please make sure they are active.

reverse navigation feature

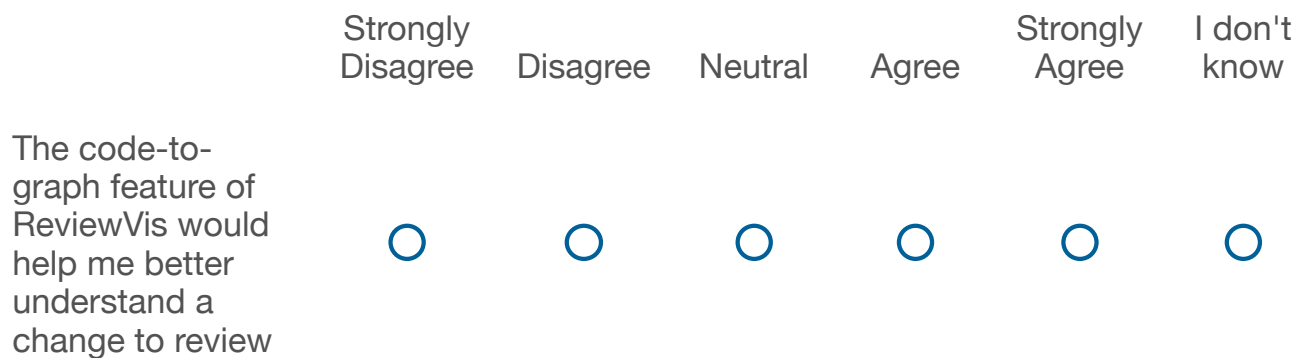


It is important that the explanation of the feature is clear. So, please answer the following question:

To which kind of files does the code-to-graph functionality apply?

- ☐ Only to Java files
- ☐ To Java and automatically generated files but not to non-Java files
- ☐ To all files (Java, automatically generated, and non-Java)

Please evaluate the following claim on ReviewVis' code-to-graph feature:



Why do you think the code-to-graph feature would support your understanding of the code to review?

Why do you think the code-to-graph feature would **not** support your understanding of the code to review?

Please evaluate the following claim on ReviewVis' code-to-graph features:



The interaction offered by this feature would support me in understanding a change to review



Block 7

In the video below, you see the graph customisation features of ReviewVis:

- Add connected nodes, i.e. class references or method calls (shift + click on node to expand)
- Remove nodes arbitrary, i.e. to track the progress on already reviewed nodes (rightclick on node to remove)

In the subtitles we will display relevant information about the feature. Please make sure they are active.

customization feature



It is important that the explanation of the feature is clear. So, please answer the following question:

Which nodes is it possible to remove from the graph?

- ☐ Only nodes representing non-Java files
- ☐ All nodes regardless of their type
- ☐ Only nodes representing methods

Please evaluate the following claim on ReviewVis' graph customisation features:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	I don't know
The graph customisation features of ReviewVis would help me better understand a change to review	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Why do you think the graph customisation features would support your understanding of the code to review?

Why do you think the graph customisation features would **not** support your understanding of the code to review?

Please evaluate the following claim on ReviewVis' graph customisation features:

Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	I don't know
-------------------	----------	---------	-------	----------------	--------------

The interaction offered by this feature would support me in understanding a change to review

<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
-----------------------	-----------------------	-----------------------	-----------------------	-----------------------	-----------------------

Block 6

Please rate the following statements about the benefits of ReviewVis:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	I don't know
Using ReviewVis would make me understand the code to review quicker	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
With ReviewVis I could orient in the code myself better while reviewing merge requests	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
With ReviewVis I would find it easier to navigate through the changes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Please rate the following statements about the perceived usefulness of ReviewVis:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Using ReviewVis in my code reviews would enable me to accomplish tasks more quickly	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Using ReviewVis
would improve my
code review
performance

☐☐☐☐☐

Using ReviewVis
in my code
reviews would
increase my
productivity

☐☐☐☐☐

Using ReviewVis
would enhance
my effectiveness
on code review

☐☐☐☐☐

Using ReviewVis
would make it
easier to do my
code reviews

☐☐☐☐☐

I would find
ReviewVis useful
in my code
reviews

☐☐☐☐☐

Please rate the following statements about the perceived ease of use of
ReviewVis:

Strongly
Disagree

Disagree

Neutral

Agree

Strongly
Agree

Learning to
operate ReviewVis
would be easy for
me

☐☐☐☐☐

I would find it easy
to get ReviewVis
to do what I want
it to do

☐☐☐☐☐

My interaction
with ReviewVis
would be clear
and
understandable

☐☐☐☐☐

It would be easy
for me to become
skillful at using
ReviewVis

☐☐☐☐☐

It would be easy
to remember how
to perform tasks
using ReviewVis

☐☐☐☐☐

I would find
ReviewVis easy to
use

☐☐☐☐☐

Block 2

What is your gender?

☐ Male

☐ Female

☐ Prefer to self-define

☐ Prefer not to disclose

Please answer to the following questions about your working experience as a developer in a professional setting; e.g. as an employee of an IT-company or as an Open-Source contributor.

	None	1 year or less	2 years	3-5 years	6-10 years	11 years or more
--	------	-------------------	---------	--------------	---------------	------------------------

For how many
years have you
developed
software in a
professional
setting?

☐☐☐☐☐☐

For how many years have you developed Java software in a professional setting?

☐ ☐ ☐ ☐ ☐ ☐

For how many years have you performed code review in a professional setting?

☐ ☐ ☐ ☐ ☐ ☐

How often do you do programming or code reviews?

Never About once a year About once a month About once a week Daily or more often

How often do you currently do programming?

☐ ☐ ☐ ☐ ☐

How often do you currently do code reviews?

☐ ☐ ☐ ☐ ☐

What is your current occupation? (multiple answers are possible)

☐ Software developer

☐ Researcher

☐ Student

☐ Other

Block 3

May we share your data (fully anonymised) in a public research dataset?

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

☐ Yes

☐ No

Did you take part in our previous survey on ReviewVis?

☐ Yes

☐ No

Do you have any final remarks about ReviewVis or this survey?

Powered by Qualtrics