

E1909 Refactor review mapping controller

From Expertiza_wiki
Jump to: [navigation](#), [search](#)

E1909. Refactoring the Review Mapping Controller[[edit](#)]

Contents

[[hide](#)]

- [1 E1909. Refactoring the Review Mapping Controller](#)
 - [1.1 About Expertiza](#)
 - [1.2 Problem Statement](#)
 - [1.3 About the Review Mapping Controller](#)
 - [1.4 Examples of Code Modification](#)
 - [1.5 Test Plan](#)
 - [1.6 Team Members](#)
 - [1.7 References](#)

About Expertiza[[edit](#)]

[Expertiza](#) is an open source project based on the [Ruby on Rails](#) framework. Expertiza provides a platform for instructors to create assignments and for students to set up teams to facilitate early communication and teamwork in these assignments. Expertiza also provides an environment where students can peer review other students, allowing these users to view feedback on their assignments in a timely manner. Expertiza supports submission across various document types, including the URLs and wiki pages.

Problem Statement[[edit](#)]

The following items were improved in the refactoring of the given file:

- Long blocks of code converted into separate methods
- Ambiguous variable names changed to meaningful variable names
- Macros added to replace hard-coded values
- Code clarity
- Test Coverage

About the Review Mapping Controller[[edit](#)]

The Review Mapping Controller handles mapping a given group's assignment submission to students working in other groups for peer review. The controller handles items associated with routing submissions to reviewers, handling permissions for instructor review calibration, and controlling drafts or uncompleted reviews. Included in this controller is the ability for the instructor to specify the number of required reviews per student or the number of reviewers per team in order to have group submission automatically sent to all students' review queues.

Examples of Code Modification[[edit](#)]

in `automatic_review_mapping` method, the following bits of code were added to separate methods:

```
participants.each do |participant|
  user = participant.user
  next if TeamsUser.team_id(assignment_id, user.id)
  team = AssignmentTeam.create_team_and_node(assignment_id)
  ApplicationController.helpers.create_team_users(user, team.id)
  teams << team
end
```

The above was moved into the `team_assignment` method that assigns users to a new team.

```
# check for exit paths first
if student_review_num == 0 and submission_review_num == 0
  flash[:error] = "Please choose either the number of reviews per student or the number of reviewers per team (student)."
elsif student_review_num != 0 and submission_review_num != 0
  flash[:error] = "Please choose either the number of reviews per student or the number of reviewers per team (student), not both."
elsif student_review_num >= teams.size
  # Exception detection: If instructor wants to assign too many reviews done
  # by each student, there will be an error msg.
  flash[:error] = "You cannot set the number of reviews done ' \
    'by each student to be greater than or equal to total number of teams ' \
    '[or "participants" if it is an individual assignment].'"
else
  # REVIEW: mapping strategy
  automatic_review_mapping_strategy(assignment_id, participants, teams, student_review_num, submission_review_num)
end
```

Again, the above is a unnecessarily large 'if' block that could be moved to another method, one that was named

validate_review_selection

The following is the final block of code moved to another method from automatic_review_mapping:

```
teams_with_calibrated_artifacts = []
teams_with_uncalibrated_artifacts = []
ReviewResponseMap.where(reviewed_object_id: assignment_id, calibrate_to: 1).each do |response_map|
  teams_with_calibrated_artifacts << AssignmentTeam.find(response_map.reviewee_id)
end
teams_with_uncalibrated_artifacts = teams - teams_with_calibrated_artifacts
# REVIEW: mapping strategy
automatic_review_mapping_strategy(assignment_id, participants, teams_with_calibrated_artifacts.shuffle!, calibrated_artifacts_num, 0)
# REVIEW: mapping strategy
# since after first mapping, participants (delete_at) will be nil
participants = AssignmentParticipant.where(parent_id: params[:id].to_i).to_a.select(&:can_review).shuffle!
automatic_review_mapping_strategy(assignment_id, participants, teams_with_uncalibrated_artifacts.shuffle!, uncalibrated_artifacts_num, 0)
```

This block was moved to the maps_strategies_for_artifacts method.

Next, the assign_reviewers_for_team method was evaluated for refactor. The following blocks of code were found that seemed to go past simply what the name of the method entailed, so they were moved to other methods:

```
participants_hash.each do |participant_id, review_num|
  participants_with_insufficient_review_num << participant_id if review_num < review_strategy.reviews_per_student
end
```

The above performs its own standalone operation, it's not needed in an already long method that has its own priorities. As such, it was moved to generate_insufficient_review_collection.

```
unsorted_teams_hash = {}

ReviewResponseMap.where(reviewed_object_id: calibrate_to: 0).each do |response_map|
  if unsorted_teams_hash.key? response_map.reviewee_id
    unsorted_teams_hash[response_map.reviewee_id] += 1
  else
    unsorted_teams_hash[response_map.reviewee_id] = 1
  end
end
```

The above creates a new hash table and performs standalone operations on it, and it is used nowhere else in the assign_reviewers_for_team method. It was moved to generate_teams_hash.

```
participants_with_insufficient_review_num.each do |participant_id|
  teams_hash.each_key do |team_id, num_review_received|
    next if TeamsUser.exists?(team_id: user_id: Participant.find(participant_id).user_id)

    ReviewResponseMap.where(reviewee_id: team_id, reviewer_id: reviewed_object_id: assignment_id).first_or_create

    teams_hash[team_id] += 1
    teams_hash = teams_hash.sort_by {|_, v| v }.to_h
    break
  end
end
```

Although the above could stay as-is in assign_reviewers_for_team, there already seems to be too much functionality in this method, which is why it was moved to insufficient_assign_reviewers

Test Plan[\[edit\]](#)

There are existing RSpec tests (spec\controllers\review_mapping_controller_spec.rb) for the given file; we were tasked with ensuring that our changes did not break any of the existing tests, thus retaining the previous code's functionality.

For the commit associated with our final submission, all existing tests pass according to travis-ci.

Team Members[\[edit\]](#)

- Randy Paluszkiwicz
- Iason Katsaros
- Weijia Li

References[\[edit\]](#)

1. [Github Project Repository Fork](#)
2. [Github Pull Request](#)
3. [Expertiza](#)

Retrieved from "https://wiki.expertiza.ncsu.edu/index.php?title=E1909_Refactor_review_mapping_controller&oldid=123088"

Navigation menu

Personal tools

- [Create account](#)
- [Log in](#)

Namespaces

- [Page](#)
- [Discussion](#)

Variants

Views

- [Read](#)
- [Edit](#)
- [View history](#)

More

Search

Navigation

- [Main page](#)
- [Recent changes](#)
- [Random page](#)
- [Help](#)

Tools

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)
- [Page information](#)

- This page was last modified on 2 April 2019, at 02:57.
- This page has been accessed 212 times.

- [Privacy policy](#)
- [About Expertiza_wiki](#)
- [Disclaimers](#)

