This page last changed on Oct 27, 2010 by Immortal.

1. Scope

1.1 Overview

The Online Review application defines a set of phase types. This component provides the plug-ins to the Phase Management component, whose logic is to check if these phases can be executed. Extra logic to execute the phases is also provided.

1.2 Logic Requirements

1.2.1 Phase Change Emails

Upon any phase change, emails should be sent to the resources associated with timeline notification for the project.

Version 1.1 already supports using different email templates for different phase starts and phase ends (although it was only used with the same email template so far). Version 1.2 further enhances this by adding the ability to configure email templates per resource role (e.g. Submitter, Reviewer, Manager etc.). This makes it possible to send different emails for submitters and managers on each phase start and phase end.

Version 1.2 also provides the ability to set default email template for all resource roles for which a specific email template is not specified. For example, a default email template can be specified for Review End. This email template won't include any review scores or other information the submitters should not see at this point. Another email template which includes the review scores will be explicitly set for Manager role and thus it will override the default template for this role. Note however that each phase start and phase end can have its own default template.
The email_samples.zip archive contains examples of the notification emails for Manager role for each phase start and phase end. All the information included in these emails will be supported by the respective phase handlers. Designers are free to add more data but at least the data included in the email samples will be supported by the respective phase handlers.This is up to the designer to decide how to reuse common fields like PHASE_TIMESTAMP and OR_LINK to avoid duplicating in each phase handler.
Also, additional configuration will be provided that will allow to turn off notification emails for particular phase start or phase end.
Designer is also responsible to provide email templates based on the email samples for each phase start and phase end. Old email template (which is common for all phases and all resource roles) is provided as a reference.
OnlineReview.xml file which contains the old configuration is also provided.

Version 1.3 was a merge from "Contest Dependency Automation" assembly and has nothing to do with the phase change emails.

Version 1.4 introduces two new phases for specification submission and review. Designers will need to provide templates for emails as well. Old templates can be reused almost entirely. The component documentation must incorporate any changes that exist in the source

Version 1.6 introduces three new phases for Milestone Submission, Milestone Screening and Milestone Review. Designers will need to provide templates for emails as well. Old templates can be reused almost entirely. The component documentation must incorporate any changes that exist in the source

1.2.2 Milestone Submission phase

This phase is similar with the existing submission phase.
The phase starts as soon as:

- the phase dependencies have been met (submission phase is started)

The phase ends:

- when the set phase end time is reached

canPerform
Determine phase status with PhasesHelper.checkPhaseStatus
If phase is "Scheduled"
Check for completed dependencies with PhasesHelper.areParentProjectsCompleted
If above conditions are met, return true. Else return false.
If phase is "Open"
Check for completed dependencies with PhasesHelper.arePhaseDependenciesMet
If above conditions are met, return true. Else return false.
perform
This method is very similar with the one for the existing submission phase.

1.2.3 Milestone Screening phase

This phase is similar with the existing screening phase.
The phase starts as soon as:

- the milestone submission phase completes

The phase ends:

- when the screening scorecards are submitted (all phases deliverables are complete)

canPerform
Determine phase status with PhasesHelper.checkPhaseStatus
If phase is "Scheduled"
Check for completed dependencies with PhasesHelper.areParentProjectsCompleted
If above conditions are met, return true. Else return false.
If phase is "Open"
Check for completed dependencies with PhasesHelper.arePhaseDependenciesMet
Check that there is at least one submission with PhasesHelper.searchActiveSubmissions
If above conditions are met, return true. Else return false.
perform
This method is very similar with the one for the existing screening phase.

1.2.4 Milestone Review phase

This phase is similar with the existing review phase.
The phase starts as soon as:

- the milestone screening phase completes

The phase ends:

- when the review scorecards are submitted (all phases deliverables are complete)

canPerform
Determine phase status with PhasesHelper.checkPhaseStatus
If phase is "Scheduled"
Check for completed dependencies with PhasesHelper.areParentProjectsCompleted
If above conditions are met, return true. Else return false.
If phase is "Open"
Check for completed dependencies with PhasesHelper.arePhaseDependenciesMet
Check that there is at least one submission with PhasesHelper.searchActiveSubmissions
If above conditions are met, return true. Else return false.
perform
This method is very similar with the one for the existing review phase.

1.2.5 Email templates

For the new review system the set of email template used for the phase end and phase start notification emails should be added for the new phases.
The old templates can be reused almost entirely.
Note that project information is available through %variable% dynamically evaluated variables (using Document Generator component).
The following information is important for the email notifications:

- for the screening email, if the submission is failed, then the reason should be included. The reason is retrieved from the only comment on the screening scorecard.

1.2.6 Local Calls

The service layer will be updated to only have local calls and no remote ejb calls. In particular that means User and Project EJBs will not longer be used as EJB but will be used as usual class instead. Concrete implementation of EJB interface will be created via reflection.
Remote calls (including web service calls) are not desired.

1.2.7 Specification Submission Phase Handler

This phase will begin when the project starts, and ends once a submission is provided.
canPerform

- Determine phase status with PhasesHelper.checkPhaseStatus
- If phase is "Scheduled"
    - Check for completed dependencies with PhasesHelper.areParentProjectsCompleted
    - If above conditions are met, return true. Else return false.
- If phase is "Open"
    - Check for completed dependencies with PhasesHelper.arePhaseDependenciesMet
    - Check that there is one submission with PhasesHelper.searchActiveSubmissions
    - If above conditions are met, return true. Else return false.

perform

- Call sendEmail

1.2.8 Specification Review Phase Handler

This phase will begin immediately after specification submission ends, and ends once review is completed.
canPerform

- Determine phase status with PhasesHelper.checkPhaseStatus
- If phase is "Scheduled"
    - Check for completed dependencies with PhasesHelper.arePhaseDependenciesMet
    - Check that there is one submission with PhasesHelper.searchActiveSubmissions whose type is for specification submissions
    - If above conditions are met, return true. Else return false.
- If phase is "Open"
    - Check for completed dependencies with PhasesHelper.arePhaseDependenciesMet
    - Get the one submission with PhasesHelper.searchActiveSubmissions whose type is for specification submissions
    - Get specification review for above submission with PhasesHelper. searchReviewsForResourceRoles for the specification review role
    - Check that review is done with review.isCommitted() == true
    - If above conditions are met, return true. Else return false.

perform

- If phase is "Open"

  - Get the one submission with PhasesHelper.searchActiveSubmissions whose type is for specification submissions
  - Get specification review for above submission with PhasesHelper. searchReviewsForResourceRoles for the specification review role
  - Check if the specification was accepted. See the phase handlers for Approval and Final Review for an example of how to do that.
  - If above conditions are met, return.
  - If here, then add another set of specification submission and review phases as stated below:
- Call sendEmail

Routine for adding new set of specification submission and review phases

- This follows very closely the steps done in PhasesHelper.insertFinalFixAndFinalReview
- Create new phases with PhasesHelper.createNewPhases
- Save phases with phaseManager.updatePhases

1.2.9 Registration Phase Handler

Registration can start as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- The period has passed;
- The number of registrations meets the required number.

When Registration phase is stopping, if there are no registrations, a Post-Mortem phase is inserted that depends on the end of the finished Registration phase.

1.2.10 Submission Phase Handler

Submission can start as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- The period has passed;
- If manual screening is absent, the number of submissions that have passed auto-screening meets the required number;
- If manual screening is required, the number of submissions that have passed manual screening meets the required number.

When Submission phase is stopping, if there are no submissions, a Post-Mortem phase is inserted that depends on the end of the finished Submission phase.

1.2.11 Screening Phase Handler

Screening can start as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- If it's primary screening mode, all submissions that passed auto-screening have one screening scorecard committed.
- If it's individual screening mode, the submission that passed auto-screening has one screening scorecard committed.

When screening is stopping, submissions with failed screening scorecard scores should be set to the status Failed Screening. The screening scores should be saved to the submitter's resource properties.When Screening phase is stopping, if no submissions have passed screening, a Post-Mortem phase is inserted that depends on the end of the finished Screening phase.

1.2.12 Review Phase Handler

Review can start as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- All active submissions have one review scorecard from each reviewer for the phase;

- All test case reviewers have one test case upload.

When Review phase is starting, all submissions failed automated screening must be set to the status Failed Screening.When Review phase is stopping, the initial scores should be aggregated and saved to the submitter's resource properties.

1.2.13 Appeals Phase Handler

Appeals can start as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- The period has passed.

1.2.14 Appeals Response Phase Handler

Appeals Response can start as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- All appeals are resolved.

When Appeals Response is stopping, all submissions with failed review scores should be set to the status Failed Review. Overall score for the passing submissions should be calculated and saved to the submitters' resource properties together with their placements. The winner and runner-up should be populated in the project properties. Submissions that do not win should be set to the status Completed Without Winning.When Appeals Response phase is stopping, if no submissions have passed review, a Post-Mortem phase is inserted that depends on the end of the finished Appeals Response phase.

1.2.15 Aggregation Phase Handler

Aggregation can start as soon as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- The winning submission must have one aggregated review scorecard committed.

When Aggregation is starting and Aggregation worksheet is not created, it should be created; otherwise it should be marked uncommitted, as well as the aggregation review comments.

1.2.16 Aggregation Review Phase Handler

Aggregation Review can start as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- The aggregation review is performed by two reviewers other than the aggregator, and the winning submitter.

When Aggregation Review phase is stopping, if the aggregation is rejected by anyone, another aggregation/aggregation review cycle is inserted.

1.2.17 Final Fix Phase Handler

Final Fix can start as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- The final fix has been uploaded;

When Final Fix is starting and Final Review worksheet is not created, it should be created; otherwise it should be marked uncommitted. Previous final fix upload will be deleted.

Updates to this phase behavior. It will be changed as per bellow implementation details.
The phase starts on demand on client request.
canPerform

- if the phase state is scheduled and not open and the project type is a studio project type verify if there is a final review scorecard already existent for the current project. Return true if it exists and false otherwise
- else perform the current business logic (no change)

perform

- if the project type is a studio project type then get the final fix scorecard for the current project, assign the winning submitter to the role of final fixes (and other step currently done with the exception of creating the scorecard and copying the comments from the aggregation scorecard)
- else perform the current business logic (no change)

1.2.18 Final Review Phase Handler

Final Review can start as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- The final review is committed by the final reviewer.

When Final Review phase is stopping, if the final review is rejected, another final fix/review cycle is inserted.When Final Review phase is stopping, if final review is approved, an Approval phase is inserted that depends on the end of the finished Final Review phase.

1.2.19 Approval Phase Handler

Approval can start as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- The approval scorecards are committed;
- All approval scorecards must have passing scores;
- At least the required number of Approver resources have filled in a scorecard (use the Reviewer Number phase criteria).

When Approval phase is stopping, if the approval is rejected, another final fix/review cycle is inserted that depends on the end of the finished Approval phase.

1.2.20 Post-Mortem Phase Handler

Post-Mortem can start as soon as the dependencies are met, and can stop when:

- The dependencies are met;
- The post-mortem scorecards are committed.
- At least the required number of Post-Mortem Reviewer resources have filled in a scorecard (use the Reviewer Number phase criteria).

1.3 Required Algorithms

No specific algorithms are required.

1.4 Example of the Software Usage

The Online Review application will plug the phases into the Phase Management component.

1.5 Future Component Direction

None.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

None.

2.1.3 Environment Requirements

- Development language: Java 1.5
- Compile target: Java 1.5

2.1.4 Package Structure

com.cronos.onlinereview.phases

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

- Database connection
- Email templates to be used for each (phase start/phase end, resource role) pair.
- Default templates to be used for each phase start and phase end.
- Whether to send notification emails on per-phase event basis.

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

JDBC

3.2.2 TopCoder Software Component Dependencies:

- Configuration Manager 2.1.5
- DB Connection Factory 1.0
- Project Phases 2.0
- Project Management 1.0.1
- Deliverable Management 1.1.1
- Phase Management 1.0.4
- Email Engine 3.1.0
- Document Generator 2.1

**Please review the TopCoder Software component catalog for existing components that can be used in the design.

3.2.3 Third Party Component, Library, or Product Dependencies:

None.

3.2.4 QA Environment:

- RedHat Linux 7.1
- Informix 10.0

3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines.

3.3.1 Database Connections

Database connections must not be cached within the component. Connections should be created for each operation and closed afterwards.

3.4 Required Documentation

3.4.1 Design Documentation

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification

3.4.2 Help / User Documentation

- Design documents must clearly define intended component usage in the 'Documentation' tab of TCUML.