Multiple evaluators workflow

This page contains macros or features from a plugin which requires a valid license.

You will need to contact your administrator.

Title:	Multiple evaluators workflow
Jira Number:	n/a
Related Jira Numbers:	n/a
Component(s):	OSP matrix / evaluations
Author:	Leidse Onderwijsinstellingen
Date:	14 jan 2008
Demo Status and Date (s):	Status: Concept Date: *(Date(s) enhancement demoed)*

Part 1: Functional Description

Multiple evaluators workflow (1)

Summary (1)

Give an overview of the envisioned enhancement, providing enough information for a non-technical user to understand what the feature is and what it would provide. Feel free to list specific features of the enhancement, but avoid implementation details and focus on functionality.

Currently OSP only supports one evaluator. Multiple (two or more) evaluators should ensure higher quality and more consistent evaluations. A workflow for two evaluators could work like this: When a student submits a matrix cell (or wizard page) for evaluation his work is evaluated by two evaluators. The first evaluator provides the second (lead) evaluator with an evaluation advice. The second evaluator creates his own evaluation based on the advice of the first evaluator and communicates the combined evaluations back to the student.

Syracuse addition

Currently OSP does not provide a means to model and route users through complex feedback and evaluation workflows. This document will outline the functionality needed to allow an Instructional Designer to represent such a workflow and to guide users through such workflows.

Indiana University addition

At Indiana University, each program determines its own evaluator policies and procedures, which ultimately dictate desired workflow. Although we have not yet received a request for evaluator routing, we have at least one program that requires two, independent blind evaluators. The cell/page should not be flagged as complete until both evaluations have been submitted. It's possible that other projects and/or programs in the future will also require two or more evaluators per cell/page, with or without routing.

Another program uses two evaluators per cell/page, but they work in teams and submit just one evaluation form per team. Teams are assigned to specific students, but currently the evaluations tool is not group aware and does not allow evaluators to be assigned on a per user or per group basis, which means the evaluations tool shows many more pending evaluations than are actually assigned to the evaluator. When we eventually implement the PUL matrix (which will be used by all undergraduate students at IUPUI), we will need much more granular control over who can see and evaluate the work of specific students and/or groups.

Rationale (1)

Explain why this feature would be valuable, and to whom. Include background information about the problem the solution is meant to solve.

Digital portfolio's are becoming more and more important in modern curricula. Portfolio's are used to measure the progress of students towards course defined competencies. Therefor students can earn credits by creating a portfolio. Accreditation organisations (for example the NVAO in The Netherlands and Flanders) demand that the evaluation process is of high quality and is consistent for different students in different courses.

Syracuse addition

The ability to model and guide teachers, students and evaluators, etc. through complex workflows will be useful for:

- Instructional Designers who want to coordinate the complex processes that will add rigor and consistency to the portfolio process and present opportunities in the process to provide the student with formative feedback.
- Evaluators and Reviewers who want the software to provide them with the information they need to quickly and easily perform the tasks required of them to author, review and assess student portfolios and make continuous, incremental improvements
- Students who wish to receive timely formative feedback about their work so that they can make continuous, incremental improvements in their work.
- Program Chairs, Deans who want to be able to oversee the portfolio processes to ensure that they are sufficiently rigorous as to meet department, college or external requirements.

Origin (1)

Describe how the need or desire for this enhancement arose, including background information about the problem it is meant to solve. Be specific about institution(s) and people who have played a role in planning the enhancement.

OSP currently doesn't support a routing mechanism to use a second evaluator in the evaluation process of a matrix cell. When using two evaluators to evaluate a student's work, they may disagree. To avoid endless arguments one evaluator plays the role of the lead /chief evaluator. S/he uses the evaluation of the first evaluator as an *advice* when performing his/her own evaluation task. It's very important that the evaluators speak in one voice to the student when they communicate the evaluation of the student's work. Thus, the student will only receive one (combined) evaluation. Communication between the evaluators doesn't nessecerily have to use Sakai tools but there should be a well defined workflow so that each evaluator knows what he has to do and when.

Syracuse addition

Syracuse University uses a portfolio assessment process that requires two evaluators to "team evaluate" a student's entire portfolio presentation (we do NOT evaluate individual cells) against our 5 criteria. The initial team of two evaluators provide a final evaluator with a evaluation recommendation with comments on each student's portfolio. The final evaluator would read the team's recommendations while viewing the student's portfolio and provide their own evaluation. This process is not modeled well in OSP.

SU evaluators look at the entire portfolio. We recognize that evidence and reflections in an integrated portfolio are likely to be found in different sections of the portfolio. Compartmentalization and scaffolding are helpful in the beginning steps of the portfolio, but as the portfolio grows and gets flushed out, our evaluators like to step back and look at the big picture. This follows a long tradition of evaluation of paper based portfolios in our school.

These user stories are nearly identical to the LOI provided stories with the obvious exception that the evaluation process starts with the publication of a portfolio for an audience of evaluators. Currently, OSP allows users to continue to add/edit/delete data from a portfolio while it is published. For a portfolio published for the purpose of evaluation, the portfolio should not be changing while the evaluators are looking at it. Rather, it should remain static...as snapshot of the student's portfolio at a moment in time (for comparison by subsequent evaluators and perhaps as a permanent record).

Indiana University addition

IUPUI's transition to teaching program follows the model described Syracuse almost exactly. One problem encountered by evaluators in this program is that evaluation forms cannot be saved in an in-progress state for later completion. The Evaluator must either save the form, inc which case it is no longer editable (nor can it be deleted from the cell/page), or abandon the process and lose any work that had been entered into the form. Ideally, the evaluator (or team of evaluators) should be able to save the form and return to it at a later time to complete it. Will the form is in this "in-progress" state it should not be visible to the matrix/wizard owner.

Another issue that has arisen at IU is the lack of distinction between cells that have never been submitted and those that have been submitted and returned to the student for revision. Currently, the cell must be reset to READY by the evaluator or the coordinator if revisions are desired. Moreover, the evaluator can only return the item to the student by competing an evaluation form and then choosing the desired status from the pulldown menu. Evaluators need to be able to return a cell/page to a student without saving evaluation data that can be seen by the student or incorporated into a report. Also, students and others need to be able to visually distinquish between cells that are *READY* (i.e. open, and never submitted) and those that have been *RETURNED*. We propose adding a fourth status for cells/pages that have been rejected and/or returned for additional work.

Finally, evaluators need an easy way to view manage multiple evaluation queues, each aggregating items at different stages in the workflow. We envision the following possibilities:
1. Items Pending Evaluation (submitted pages/cells that have not yet been claimed by an authorized evaluator)
2. In-Progress Evaluations (evaluations that have been started, but not completed, by the current evaluator)
3. Resubmissions (cells/pages that have been returned to the student for revision by the current evaluator, revised by the student, and then resubmitted.)

Additional details on how these changes might be realized are found in the attachment, IU evaluator requirements and workflow.15Feb2007.pdf

User Stories (1)

The User Stories should paint a picture of what it is like for a user to make use of the enhancement. The actors should be based on real users with definable tasks and goals. Include as many stories as necessary to demonstrate how the enhancement would be used by different types of users.

Actors and Stakeholders

- Actor: Student
- Actor: Evaluator 1
- Actor: Evaluator 2 (lead/chief evaluator)
- Stakeholder: Educational institute (high quality, consistent evaluations)

User story 1: Both evaluators approve

1. Student submits his cell for evaluation.
2. Evaluator 1 receives a notification with an evaluation request.
3. Evaluator 1 fills out an evaluation based on student's work. Student's work looks examplary.
4. Evaluator 1 selects next workflow step: *approve and continue evaluation*
5. Evaluator 2 receives a notification with an evaluation request.
6. Evaluator 2 reads evaluator 1's evaluaton
7. Evaluator 2 fills out an evaluation based on student's work and evaluator 1's evaluation. Student's work looks examplary and evaluator 1's evaluation is appropriate.
8. Evaluator 2 selects next workflow step: *approve and return to student*

9. Student receives a notification
10. Student reads combined evaluation from evaluator 1+2

User story 2: First evaluator cannot complete evaluation

1. Student submits his cell for evaluation
2. Evaluator 1 receives a notification with an evaluation request
3. Evaluator 1 sees that critical evidence is missing and that he cannot complete his evaluation
4. Evaluator 1 writes an evaluation in which he indicates the missing items
5. Evaluator 1 selects the next workflow step: *reject and return to student*
6. Student receives a notification
7. Cell/page status is set to "Returned" (IU Addition)
8. Student reads evaluation from evaluator 1

User story 3: First evaluator approves, second evaluator rejects

1. Student submits his cell for evaluation
2. Evaluator 1 receives a notification with an evaluation request
3. Evaluator 1 fills out an evaluation based on student's work. Student's work looks ok.
4. Evaluator 1 writes an evaluation
5. Evaluator 1 selects the next workflow step: *approve and continue evaluation*
6. Evaluator 2 receives a notification with an evaluation request.
7. Evaluator 2 reads evaluator 1's evaluation
8. Evaluator 2 fills out an evaluation based on student's work and evaluator 1's evaluation. Evaluator 2 believes student's work still needs some work.
9. Evaluator 2 selects next workflow step: *reject and return to student*
10. Student receives a notification
11. Cell/page status is set to "Returned" (IU Addition)
12. Student reads combined evaluation from evaluator 1+2

User story 4: First evaluator rejects, second evaluator approves

1. Student submits his cell for evaluation
2. Evaluator 1 receives a notification with an evaluation request
3. Evaluator 1 fills out an evaluation based on student's work. Evaluator 1 beleives student's work is below expectation.
4. Evaluator 1 writes an evaluation
5. Evaluator 1 selects the next workflow step: *reject but continue evaluation*
6. Evaluator 2 receives a notification with an evaluation request.
7. Evaluator 2 reads evaluator 1's evaluation
8. Evaluator 2 fills out an evaluation based on student's work and evaluator 1's evaluation. Evaluator 2 believes student's work is ok.
9. Evaluator 2 contacts evaluator 1 to discuss. After discussion both evaluators agree student's work meets expectations.
10. Evaluator 2 selects next workflow step: *aprove and return to student*
11. Student receives a notification
12. Student reads combined evaluation from evaluator 1+2

User story 5: Two step evaluation of a published portfolio - Syracuse (Feb 2, 2008)

(See SAK-10529)

1. Student publishes a portfolio for evaluation
2. Evaluator 1 receives a notification with an evaluation request.
3. Evaluator 1 fills out an evaluation based on student's work. Student's work looks examplary.
4. Evaluator 1 selects next workflow step: *approve and continue evaluation*
5. Evaluator 2 receives a notification with an evaluation request.
6. Evaluator 2 reads evaluator 1's evaluaton
7. Evaluator 2 fills out an evaluation based on student's work and evaluator 1's evaluation. Student's work looks examplary and evaluator 1's evaluation is appropriate.
8. Evaluator 2 selects next workflow step: *approve and return to student*
9. Student receives a notification
10. Student reads combined evaluation from evaluator 1+2

User story 6: First evaluator cannot complete evaluation of a published portfolio - Syracuse (Feb 2, 2008)

(See SAK-10529)

1. Student publishes a portfolio for evaluation
2. Evaluator 1 receives a notification with an evaluation request
3. Evaluator 1 looks at the previous evaluation data and previous versions of the portfolio and this published portfolio
4. Evaluator 1 sees that critical evidence is missing and that he cannot complete his evaluation
5. Evaluator 1 writes an evaluation in which he indicates the missing items
6. Evaluator 1 selects the next workflow step: *reject and return to student*
7. Student receives a notification
8. Cell/page status is set to "Returned" (IU Addition)
9. Student reads evaluation from evaluator 1

User story 7: First evaluator approves, second evaluator rejects a published portfolio - Syracuse (Feb 2, 2008)

(See SAK-10529)

1. Student publishes a portfolio for evaluation
2. Evaluator 1 receives a notification with an evaluation request
3. Evaluator 1 looks at the previous evaluation data and previous versions of the portfolio and this published portfolio
4. Evaluator 1 fills out an evaluation based on student's work. Student's work looks ok.
5. Evaluator 1 writes an evaluation
6. Evaluator 1 selects the next workflow step: *approve and continue evaluation*
7. Evaluator 2 receives a notification with an evaluation request.
8. Evaluator 2 reads evaluator 1's evaluation
9. Evaluator 2 fills out an evaluation based on student's work and evaluator 1's evaluation. Evaluator 2 believes student's work still needs some work.
10. Evaluator 2 selects next workflow step: *reject and return to student*
11. Student receives a notification
12. Cell/page status is set to "Returned" (IU Addition)
13. Student reads combined evaluation from evaluator 1+2

User story 8: First evaluator rejects, second evaluator approves a published portfolio - Syracuse (Feb 2, 2008)

(See SAK-10529)

1. Student publishes a portfolio for evaluation
2. Evaluator 1 receives a notification with an evaluation request
3. Evaluator 1 looks at the previous evaluation data and previous versions of the portfolio and this published portfolio
4. Evaluator 1 fills out an evaluation based on student's work. Evaluator 1 beleives student's work is below expectation.
5. Evaluator 1 writes an evaluation
6. Evaluator 1 selects the next workflow step: *reject but continue evaluation*
7. Evaluator 2 receives a notification with an evaluation request.
8. Evaluator 2 reads evaluator 1's evaluation
9. Evaluator 2 fills out an evaluation based on student's work and evaluator 1's evaluation. Evaluator 2 believes student's work is ok.
10. Evaluator 2 contacts evaluator 1 to discuss. After discussion both evaluators agree student's work meets expectations.
11. Evaluator 2 selects next workflow step: *aprove and return to student*
12. Student receives a notification
13. Student reads combined evaluation from evaluator 1+2

User Story 9: Different evaluators with different evaluation devices/forms - *UMich (Feb 2, 2008)*

This user story is for the Undergraduate Research Opportunities Program. In this program, students are evaluated by both their faculty mentor and the peer advisor who runs a weekly seminar.

1. Student submits matrix cell for evaluation
2. Evaluator 1 (peer adviser) fills out peer-advisor-form, which asks about things like the student's attendance and performance in the peer-advisor-led seminar
3. Evaluator 2 (faculty mentor role) reads peer advisor evaluation and takes it into account when filling out faculty-mentor-form. The faculty mentor's form covers all aspects of the program, both participation in the seminars (which the faculty mentor knows about through the peer advisor evaluation) and research work done directly under the faculty mentor.
4. Peer adviser meets with student to discuss both evaluations
5. After this meeting the peer adviser unlocks the cell, making the two evaluations available to the student.

User Story 10: Two or More Evaluations Required- No Routing - IU (Feb 15, 2008)

1. Matrix/wizard administrator sets required number of evaluators per cell/page to 2.
2. Matrix/wizard admin selects evaluators (number of evaluators equals or exceeds number of required evaluations)
3. Student submits matrix/wizard cell/page X for evaluation
4. Evaluators who have opted in for individual or digest notifications, receive an email notification.
5. Evaluator 1 logs in to evaluator dashboard and selects the cell/page X.
6. Evaluator 1 writes and submits evaluation for cell/page X.
7. Cell/page X status is not set to complete because evaluation count has not met requirement in step 1.
8. Evaluator 2 logs in to evaluator dashboard.
9. Evaluator 2 select cell/page X for evaluation.
10. Cell/page X disappears from the dashboard of all approved evaluators for this cell/page because quota will be met when Evaluator 2's work is complete.
11. Evaluator 2 writes and submits evaluation cell/page X.
12. Status of cell/page X is set to complete.
13. Student receives email notification.

User Story 11: Blind Evaluation - IU (Feb 15, 2008)

1. Matrix/wizard administrator sets required number of evaluators per cell/page to 1.
2. Matrix/wizard admin selects evaluators for cell/page X.
3. Student submits matrix/wizard cell/page X for evaluation

4. Evaluators who have opted in for individual or digest notifications, receive an email notification.
5. Evaluator 1 logs in to evaluator dashboard and selects the cell/page X. Identity of student is not visible. (Evaluator cannot access cell/page X directly via the username dropdown in the parent matrix or wizard. This is critical for blind evaluation)
6. Cell/page X disappears from the dashboard of all approved evaluators because quota will be met when Evaluator 1's work is complete.
7. Evaluator 1 writes and submits evaluation for cell/page X.
8. Status of cell/page X is set to complete.
9. Student receives email notification.

User Story 12: Evaluator Saves "In-Progress" Evaluation Form for Later Editing - IU (Feb 15, 2008)

1. Matrix/wizard administrator sets required number of evaluators per cell/page to 1.
2. Matrix/wizard admin selects evaluators for cell/page X.
3. Student submits matrix/wizard cell/page X for evaluation
4. Evaluators who have opted in for individual or digest notifications, receive an email notification.
5. Evaluator 1 logs in to evaluator dashboard and selects the cell/page X.
6. Cell/page X disappears from the dashboard of all other approved evaluators because quota will be met when Evaluator 1's work is complete.
7. Evaluator 1 starts filling out evaluation form for cell/page X, but is interrupted before completing it.
8. Evaluator 1 saves in-progress form for cell/page X.
9. Cell/page X is moved to Evaluator 1's pending/in-progress evaluation queue.
10. Evaluator 1 logs out.
11. Evaluator 1 logs in at a later time.
12. Evaluator 1 selects cell/page X from his/her pending/in-progress evaluation queue in dashboard.
13. Evaluator 1 completes in-progress evaluation for cell/page X and submits.
14. Status of cell/page X is set to complete.
15. Student receives email notification.

Note: if in-progress evaluations are not completed by Evaluator 1 within *n* days, the matrix/wizard admin is notified.

User Story 13: Evaluator Returns Cell/Page to Student for Revisions - IU (Feb 15, 2008)

1. Matrix/wizard administrator sets required number of evaluators per cell/page to 1.
2. Matrix/wizard admin selects evaluators for cell/page X.
3. Student submits matrix/wizard cell/page X for evaluation
4. Evaluators who have opted in for individual or digest notifications, receive an email notification.
5. Evaluator 1 logs in to evaluator dashboard and selects the cell/page X.
6. Cell/page X disappears from the dashboard of all approved evaluators because quota will be met when Evaluator 1's work is complete.
7. Evaluator 1 chooses *return for revisions* to request addition work for student.
8. Status of cell/page X is set to *Returned*.
9. Student receives email notification.
10. Student revises cell/page X and resubmits.
11. Evaluator 1 is notified of resubmission.
12. Cell/page X is placed in Evaluator 1's *Review and Complete Resubmissions* queue.
13. Evaluator 1 logs in at a later time.
14. Evaluator 1 selects cell/page X from his/her *Review and Complete Resubmissions* queue.
15. Evaluator 1 writes and submits an evaluation for cell/page X.
16. Status of cell/page X is set to complete.
17. Student receives email notification.

Note: if evaluations for resubmitted items are not completed by Evaluator 1 within *n* days, the matrix/wizard admin is notified.

User Story 14: Evaluator Forwards Cell/Page to Admin for Reassignment- IU (Feb 15, 2008)

1. Matrix/wizard administrator sets required number of evaluators per cell/page to 1.
2. Matrix/wizard admin selects evaluators for cell/page X.
3. Student submits matrix/wizard cell/page X for evaluation
4. Evaluators who have opted in for individual or digest notifications, receive an email notification.
5. Evaluator 1 logs in to evaluator dashboard and selects the cell/page X.
6. Cell/page X disappears from the dashboard of all approved evaluators because quota will be met when Evaluator 1's work is complete.
7. At some point (the evaluation may be *in-progress* or *resubmitted)*, Evaluator 1 determines that s/he is unable to complete the cell/page X evaluation or has a question for the matrix/wizard admin.
8. The Evaluator forwards cell/page X to the matrix/wizard admin.
9. A notification is sent to the matrix/wizard admin.
10. The matrix/wizard admin logs in a locates the forwarded item in his *Forwarded Evaluations* queue.
11. The matrix/wizard admin either answers question and returns cell/page X to Evaluator 1 for completion or (if Evalluator 1 cannot complete), releases cell/page X back into the Itmes Pending Evaluation queue.

User Story 15: Multiple Evaluators Use Evaluation Form with Clickable Descriptors to Rate Criteria and Goals and Calculate Average Ratings - rSmart (March 24, 2008)

A) Functionality that already exists in 2.5 (with customization of forms).

1. An evaluation form that uses .xslt to provide a scoring rubric in the form of matrix with clickable descriptors is associated with a matrix cell, wizard page. The evaluation form provides rows for each criterion (standard, outcome, etc.) and columns for each scoring level.
2. The cells in the matrix provide descriptors for student performance for each criterion at each scoring level. Evaluators click on one matrix cell per row to indicate the score they award to each participant for each criterion. The cell that has been selected as the score for each row is highlighted for visual confirmation that it has been selected.
3. Each time an evaluation form is submitted by an evaluator, the mean score across all criteria is calculated and displayed on the form for viewing by any user with permission.
4. The evaluation form may provide a comment area for evaluators to include comments.

B) Functionality desired for OSP 2.6 or beyond (with customization of forms).

1. Evaluation forms are also used to rate goals linked to a matrix cell or wizard page via the goal management tool.
2. Evaluation forms, specified evaluators, and the goal management process may also be used with portfolios via the association of evaluation forms with portfolio templates.
3. When evaluators add comments to an evaluation form instance, they may designate them as public or private comments.
4. When each evaluator submits an instance of an evaluation form for a matrix cell, wizard page, or portfolio, the calculations for each criterion or goal in that form instance, as well as the mean score for the form instance, are combined with calculations from all other evaluation form instances for the owner of that cell, page, or portfolio to provide a mean score for each criterion or goal and a total mean score across all evaluators.
5. The set of evaluators associated with a matrix cell, wizard page, or portfolio template can include one or more peer advisers whose ratings are advisory only and do not enter into the calculation of mean scores.
6. The calculation of the mean score for each criterion or goal identifies an acceptable range of inter-rater reliability for each rating and flags ratings that require a third evaluator.
7. A flagged evaluation form instance requires the submission of an additional evaluation form instance by a third evaluator before the cell, page, or portfolio can be complete. Ratings from the additional evaluator replace the scores from the evaluator farthest from the calculated mean score for the cell, page, or portfolio.
8. When all evaluators have completed their ratings, the cell, page, or portfolio becomes complete and its contents and score cannot be changed.
9. The owner of a completed matrix cell, wizard page, or portfolio can view combined ratings and public comments of all evaluators.

Functional Details (may be added after community demo) (1)

See: Multiple Evaluator Workflow - Functional design

Describe any functionality not fully captured in the User Stories.

Interaction and Implications (1)

- OSP Matrix tool
- OSP Wizard Tool
- OSP Evaluation tool
- Assignments tool
- (future) Workflow tool?

Diagrams and Mockups (3)

Include any ERDs, flowcharts, sketches, mockups, etc.

Community Acceptance (4)

Indicate how this feature has been discussed by the larger community (e.g., list discussion, specific meetings, etc.). Provide specific records of community acceptance (e.g., list institutions and contacts who also identify this feature as a requirement).

This enhancement has been discussed on the weekly OSP teleconf. calls.

Institutions who identify this feature as a requirement:

- LOI
- University of Michigan (use cases pending)
- rSmart (use cases pending)
- Indiana University (use cases pending)

(Please add your institution here if you support this enhancement)

Part 2 of the Proposal for Enhancement Template: The Specification*

The specification should be filled out once the feature is clearly defined.

Specification Template (5)

Behavior

Describe each specific behavior of the feature in the present tense as if the feature were implemented perfectly. Use precise, objective language to describe ideal behaviors against which actual behaviors can be evaluated.

In the case of conditions and behaviors that must be evaluated independently, they should be presented in a two-column table as below.

Conditions	Behavior
(Short description of mutually exclusive condition #1)	*(Objective, verifiable behavior in response to condition #1)*
(Short description of mutually exclusive condition #2)	*(Objective, verifiable behavior in response to condition #2)*

When there are workflow behaviors (steps) that must be evaluated in sequence, they should be identified with prerequisite conditions, behavior, and post-behavior conditions as below.

Workflow Steps

(Unique, short, representative name of the step)	
Prerequisite Conditions or Step:	*(Conditions or Step name)*
Behavior:	*(Objective, verifiable behavior)*
Post-step Conditions or Next Step:	*(Conditions or Step name)*

Interaction

List any entities or actors that are used or affected by this feature. Each should link to an entry in the OSP Terminology page.

Quality Metrics

Describe any non-functional requirements of the feature, such as usability, performance, or design. Provide objective and, where possible, quantitative measures against which actual implementations can be evaluated.

Assumptions

Provide any assumptions about implementation path, availability of other required features, schedule concerns or otherwise.

Outstanding Issues

The Outstanding Issues section is a placeholder for the evolution of this specific feature. It should mention any explicit design or implementation decisions that are pending. There must be no outstanding decisions as of the confirmation of the feature as a requirement.