Team LMNO

Project: RIT Co-op Evaluation System

Jonathon Leight

Trevin Maerten

Noel Nacion

John O’Connor

**Table of Contents**

[1. Overview](#h.k00v22fumz5x)

[2. Goals and Scope](#h.tbku1yionqa7)

[3. Deliverables](#h.o21ihgan1tto)

[4. Risk Management](#h.3a78c5htedi2)

[5. Scheduling and Estimates](#h.nxr6vkg362q7)

[Work Breakdown](#h.l4xka748b4w3)

[Overall Schedule](#h.j4ylqm7h9b8s)

[First Quarter](#h.ki13pnwujjug)

[Second Quarter](#h.svhb3irxyqcy)

[End of Second Quarter](#h.oq5fj8msrsnd)

[Overall](#h.5epqxxinux30)

[6. Measurements & Metrics](#h.n10w0o63ek71)

[7. Technical Process](#h.ke0wnf7xhfz)

# 1. Overview

The RIT Co-op Evaluation system is a system by which students and employers can submit evaluations about their mutual co-op experience. The system is accessed via a browser. Users must authenticate themselves before they will be given access to the system. Once authenticated, the users can fill out and submit any evaluations that they are authorized to fill out. The evaluations can be saved and edited at a later date for convenience. The evaluation system also allows certain authorized users to review the submissions and preform analysis on collections of submissions. Notifications can be sent by the system to its users to remind them to fill out the evaluations when the deadline for submissions approaches.

There are several user classes that would be using the RIT co-op evaluation system. First there are the students who go on co-op. They need to use the system in order to submit their own evaluation of their experiences on their co-op. This submission is a necessary part of their grade. Secondly there are the employers of the co-oping students. They too have to complete and submit a evaluation of the student. Thirdly are the RIT faculty which are given authorization to access co-op submissions for students and employers. Access to the submissions are restricted to those within the same college as the student. The faculty can also manually send out notifications to either students or employers to remind them to complete the evaluation. The ability to aggregate the data is also given to the RIT faculty. Lastly there are the system administrators which are given extra privileges in order to maintain the system.

The original RIT Co-op Evaluation system was built using Java and Oracle. Over time development teams have added onto the initial functionality as the user’s needs changed. The current system is still built upon the same technologies that were used in its original design. The new proposed system will be designed from scratch in order to make use of newer, better technologies including C# and MVC3 with MSSQL.

Ideally the development work for this project will be completed before the end of the 20 week session allocated. Further details on the schedule will be found in the later section. Besides the hard deadline at the end, there will be periodical checkups for the sponsor in order to communicate the current status of the project and what the next steps will be. It will be the team’s responsibility to complete the agreed upon work for each checkpoint and provide a stable, working product at the end of the session.

# 2. Goals and Scope

LMNO will be re-engineering the existing Co-op Evaluation System. The scope will be equal to that of the system that is currently in place without additional functionality. Changing the functionality of some of the components of the old system, and increasing the usability of some of the components will be in scope. Configuring the authentication to work with the RIT LDAP, migrating the data from the old into the new database, re-engineering the database, and sections listed in the work breakdown schedule will be in scope for our final release. LMNO will focus its concerns on usability and performance to create a more pleasurable experience. The reports, when submitted, will fail less than 5% of the time and will experience a successful submission within 1 minute during normal working conditions. Editing reports within the browser window will be possible.

Setting up the hosting environment for the evaluation system will not be in scope for the development team. Neither will be any integration of the evaluation system with any existing or future systems.

# 3. Deliverables

* Create team account providing: team name, login-ids of each team member and faculty coach
* Submit project synopsis of no more than 250 words to sponsor and faculty coach for approval
* Project website on an se.rit.edu machine
* URL for the project website
* Publish, on project website, team information including approved synopsis
* Publish, on project website, first tracking report of time/effort worked
* Complete project information survey form
* Submit first draft of project plan for review by faculty coach and sponsor
* Document development methodology on project website
* Document product/process metrics on project website
* Make interim status and final project presentations.
* Attend presentations given by other teams and provide constructive feedback on the presentations
* Project poster
* Project presentation
* Technical report outline, followed by project technical report
* Interim and final team self-assessment
* Post-mortem curriculum reflection report
* Final working code to project sponsor
* CD(s) at the conclusion of the project containing all project artifacts
* Software Engineering Program Senior survey

# 4. Risk Management

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Description** | **Probability** | **Cost (hours)** | **RE** | **Mitigation Strategy** |
| Virtual machine not available | .05 | 50 | 2.5 | Save local copies as emergency back-ups |
| Sponsor not available for meeting | .1 | 20 | 2 | Plan out 2 weeks of work in advance |
| Team members clash | .1 | 50 | 5 | Appoint team leader to help resolve conflicts |
| Disagreement with project sponsor | .4 | 30 | 12 | Engage in discussions when disagreement arises and resolve quickly |
| Team member scheduling conflicts | .25 | 50 | 12.5 | Plan meetings in advance and during specified Senior Project times (TR 4-6) |
| Unfamiliarity with programming language | .5 | 120 | 60 | Use online tutorials and resources to quickly learn the language |
| Unfamiliarity with project tools | .5 | 80 | 40 | Use documentation and resources to quickly learn new tools |
| Unfamiliarity with current Co-op evaluation system | .25 | 40 | 10 | Navigate through the current system to learn it’s features |
| Project coach unavailable for meeting | .1 | 40 | 4 | Plan meetings in advance or communicate via e-mail |
| Incorrect analysis of requirements | .05 | 80 | 4 | Create list of requirements to be verified by the project sponsor |
| Unknown risks | .5 | 10 | 5 | Plan ahead and don’t fall behind schedule |

# 5. Scheduling and Estimates

## Work Breakdown

**Co-op Evaluation System Components**

* View reports: searching by student, searching by college, searching by date, displaying report in the browser window.
* Student report: filling out and submitting the report, viewing the report (submitter only)
* Employer report: filling out and submitting the report, viewing the report (submitter only)
* Notifications: send, check status
* Admin tasks: editing reports, adding / removing users, changing privileges, transfer students, paper version submission
* Utilities: import evaluations, change status

## Overall Schedule

### First Quarter

* Create mock-ups (end of week 8)
* Interview users (students and staff) for understanding of work-flow (end of week 10)
* Familiarize with the existing system (end of week 4)
* Familiarize with C# (end of week 8)
* Familiarize with development tools and environment (end of week 8)
* Create requirements document for the system (end of week 6)
* Create architecture document for the system (end of week 10)

### Second Quarter

* Finish agreed upon functionality for a working system (end of week 4)
* Focus on usability and accessibility for an improved user experience (end of week 8)

### End of Second Quarter

* Deliver final project to sponsor (end of week 8)
* Create poster and hold a presentation during Senior Project Day (end of week 10)

### Overall

* Weekly meetings with project sponsor (Thursdays from 4-6)
* Bi-weekly updating of time tracking
* Bi-weekly meetings with project coach (Wednesdays)
* Other, currently unscheduled, team meetings used for collaboration and for getting work done

# 6. Measurements & Metrics

Two metrics will be tracked throughout the project in order to help keep track of the team’s progress.

1. Time spent: will be tracked in order to make sure that the team is doing the work they are supposed to be doing, and in order to help determine possible problems with the schedule.
2. Bugs found: the number of bugs found will help to determine how much more work will need to be done on the project before it can be released in a working state.

These metrics were chosen due to the team’s unfamiliarity with the programming language, libraries, and the tools that have been provided for the team to use, as well as the restrictions placed on the team. Given these metrics, the team believes it will be easier to determine the progress of the project, as well as making it easier to predict how long each requirement will take to implement.

# 7. Technical Process

LMNO has determined to use the spiral methodology. This will allow LMNO to check with the project sponsor to ensure correct interpretation of the requirements and a steady development process. It will also allow the project sponsor to gauge progress on the overall project. The length of each spiral will be two weeks and will consist of all of the deliverables that are required for those two weeks.

LMNO will be completing the project using the following tools:

* C# - Development language that is currently used by the RIT co-op office
* Visual Studio 2010 Professional - IDE for C# development
* Microsoft SQL Server 2008 - relational database management system
* ASP.NET MVC3 - web development framework for C#
* PetaPoco - ORM framework
* ELMAH - error handling and logging modules
* Foolproof Validation - extra validation annotations
* Data Annotations Extensions - extra validation annotations
* NUnit - unit testing framework
* Ninject - dependency injection framework
* jQuery - javascript framework
* Mercurial - distributed version control system