Redmine as a Web-Based Collaboration Tool in Engineering Design Courses

Prof. Junichi Kanai, Rensselaer Polytechnic Institute

Junichi Kanai received a B.S. in EE, and a Master of Engineering and a Ph.D. in CSE from RPI (Rensselaer Polytechnic Institute) in 1983, 1985, and 1990, respectively. He was an Assistant Research Professor at the Information Science Research Institute, University of Nevada, Las Vegas, from 1990 to 1997. Dr. Kanai joined Panasonic Information and Networking Technologies Lab in Princeton, NJ in 1998. He was a senior scientist developing and transferring new technologies to Panasonic product divisions in Japan. He was also responsible for managing his groups' patent portfolio. From 2002 to 2004, he was a manager at the system group of Panasonic's sales company in Secaucus, NJ providing system integration and software development for clients. He was also an Export Control officer. Dr. Kanai joined the Design Lab at RPI in 2004. He is currently the Associate Director of the lab and and Professor of Practice of in the Electrical, Computer, and Systems Engineering department. The Design Lab provides industry sponsored and service oriented multidisciplinary design projects to 200 students/semester. His responsibilities include managing the operation of the Design Lab and enhancing the experience for students working on engineering design projects.

Mark Anderson, Rensselaer

He received his Bachelor's degree and his Master's degree in Electrical Engineering from Rensselaer Polytechnic Institute in 1979 and 1980, respectively. Mark began his career at Rensselaer as a Project Engineer for the Center for Manufacturing Productivity and Technology Transfer. He then moved to Westinghouse and General Electric where he held various positions in several computer related fields including manufacturing, robotics, vision systems, instrumentation, software development, factory automation, engineering, numerical control/CNC, machining, and business analysis. Mark is currently a Senior Project Engineer for the Design Lab at Rensselaer and the Course Coordinator for the Introduction to Engineering Design class.

Redmine as a Web-Based Collaboration Tool
in Engineering Design Courses

Abstract

An open source web-based collaboration system called Redmine was introduced for use in our capstone design program. Many project teams effectively used both the project's wiki, discussion forums, and a repository tool for recording and communicating their design work. The students also learned to use the discussion forums and the issue tool to manage their projects. On the other hand, various user statistics obtained from the system allowed the faculty advisors to monitor the frequency of each student's contributions and to quickly review the content and quality. The system made a significant impact on the outcome of the project results. This paper will present issues in deploying the tools, the best practices for using these tools, and assessing students' performance in capstone design courses.

1. Introduction

To become successful engineers, students must learn technical knowledge, good communication, skills, and teamwork skills. Traditional lecture-based coursework focuses on providing a solid theoretical foundation and analytical skills for each of the various disciplines. On the other hand, laboratory courses and engineering design courses are often used to teach communication and teamwork skills [1]. Typical communication skills include, but are not limited to, maintaining lab/design notebooks, writing technical reports, and oral presentations. A project-based course may also include writing a proposal.

On-line collaboration tools, also known as groupware, are widely used in many organizations to improve their productivity and the quality of their products. Currently, Wikipedia includes over 95 software tools [2]. Types of collaboration tools include bulletin (discussion) boards for threaded discussions, public folders for sharing documents, and version control systems for concurrent editing software source codes or CAD files. Therefore, it is beneficial for students to learn the skills to use such tools in engineering projects.

In our capstone design program, students work with their academic advisors as well as industry sponsors who are typically out of town, toward a common project objective. An open source web-based bulletin board system was introduced for enhancing the design program in the spring of 2005 and made a significant impact on the outcome of the project results [2]. Although the system provided many benefits to our program, better collaboration tools have since been developed. Hence, we migrated from the bulletin board system to an open source bug tracking system called Redmine [5] in the fall of 2010. Redmine was developed by Jean-Philippe Lang and contributors.

Learning to use the system and manage a design project effectively also contributes to the following ABET student outcomes [3]:

- (k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.
- (g) an ability to communicate effectively
- (d) an ability to function on multidisciplinary teams

In the following sections, we will present issues in deploying the system and the best practices for using these tools in capstone design courses.

2. Background and Problems

In our capstone design courses, a multidisciplinary team of senior engineering students works on a problem specified by an industrial partner that is typically a global company. Workers in such a company require skills to collaborate with peers among multiple locations. Since many of our students join global organizations, our capstone design emphasizes team, communication, and skills to use collaboration tools.

Currently a typical team consists of seven students from multiple programs, such as Biomedical Engineering, Computer System Engineering, Electrical Engineering, Industrial Management Engineering, Material Engineering, and Mechanical Engineering. Since students take an introductory engineering design course that also includes teamwork and project management skills in their sophomore year, the capstone design course is a semester long (15 weeks) and does not have any lecture components. It typically takes two or more semesters to complete a project. A sample sequence is as follows:

- First Semester – Identify requirements and build a proof of concept
- Second Semester – Design and build the end-to-end prototype
- Third Semester – Re-design and build a field testable prototype
- Fourth Semester – Refining and iterating the prototype

Because of the nature of our capstone design program, our students face the following challenges:

- Students spend 7% (1/15 of a semester) of their time budget every week. It is extremely important for them to be well organized and execute necessary tasks in a timely manner.
- Students need to collaborate with not only their faculty advisors but also their sponsor mentors who are, in general, out of town. Moreover, in the fall of 2012, 25% of the projects involved mentors from multiple locations, and one project involved mentors from overseas.
- All design artifacts must be made available to future teams. In the fall of 2012, 90% of the projects were multisemester.

3. Selecting a System

A variety of collaboration tools are widely available. Functions useful in engineering design courses include:

- Record keeping, such as Wiki and discussion board supporting attachments
- Technical discussions and design reviews, such as discussion board
- Revision control of design documents, such as a version control system
- Project/Task management, such a Gantt chart tool
- User statistics (log capability) useful in evaluating a student's performance

Since each university has a unique set of computer/network policies and support capabilities, there is no readymade recommendation that fits every school's needs. Key IT issues in selecting a system are as follows.

- Technical supports available on campus: IT groups are typically comfortable with either Microsoft or Unix/Linux technologies. Successful deployment of a system often depends on the availability of technical support.
- Budget: The total cost for deploying a system includes hardware, software licenses, and ongoing support efforts.
- User Account Management: - It is easier for both students and a system administrator to use/manage tools that are accessible by a single account. If a project involves external users, such as sponsor mentors, allowing them to access the system might be an issue.
- Back Up and Availability: Once students start using the system, it must be available until the end of a semester and the information must be protected by regular back ups.

Our experience with these issues in setting up our first bulletin board based collaboration system was reported in the 2008 ASEE Zone 1 Conference [4]. The following sections will describe some of these issues addressed in the migration process.

4. Redmine in Our School

The bulletin board system allowed students to learn and practice electronic record keeping, on-going design review, and revision control [4]. However, it did not provide any project management functions. Therefore, we looked for another system that also provides project management related functions.

Our first installation of Redmine [5] (version 0.8.x in the summer of 2009) was used for our initial prototyping and experimentation. The Redmine package is extremely powerful and very configurable. As a result, it took a yearlong pilot project to find the best way for setting up the system to meet our needs. Some of these decisions resulted in specific Redmine configuration while other settings were addressed by establishing conventions for system usage.

4.1 Basic Workflow Settings and Security Policies

Our operation of Redmine is intended to serve as the project's design notebook, in a manner similar to traditional paper log books. The students record design documentation in the 'issues' and 'forums'. Our security settings allow the students to use most of the capabilities except for deleting material - just as one typically would not erase or delete material from a paper design

notebook. This practice also applies to software, CAD files, and other documentation by using Subversion or a similar change control tool. While students can mark things as deleted in the repository, the change control software does not actually delete things. Material can be readily retrieved whether accidentally or intentionally marked for deletion. By maintaining the integrity of the design data students learn the importance of complete and accurate record keeping.

4.2 Features Enabled for Users

Redmine's security model also supports the concept of 'roles' and 'responsibilities'. Each defined role has an associated set of responsibilities for each project. These responsibilities translate to the workflow and security settings of Redmine. For our students, we have created the role of 'Team Member'. Team Members can add or edit wiki pages, create and update issues, add threads in the forums and reply to the posts of their team mates and can work with the files in the change control repository.

To acknowledge and help identify our sponsors, they are assigned the role of 'Sponsor'. Our faculty participates in the program as evaluators with the role of 'Chief Engineer' while our staff members who provide the day to day project management are assigned the 'Project Engineer' role. For our purposes, these roles have very similar security settings with slightly more control going to the Project Engineers. Our Project Engineers have the additional capabilities necessary for managing their projects such as configuring and maintaining the project's wiki and other Redmine details.

4.3 Features Enabled for Administrators

A number of features and functions are reserved for our system administrators. These include the typical security and system wide configuration management utilities and the ability to add and delete projects.

Our approach is to create a separate project in Redmine for each student team. This allows for the segregation of intellectual property and a coherent organization of project information. For projects that continue across multiple semesters, students continue to work within the same project space. This allows the teams to access the design artifacts left by the previous teams.

Each project has a dedicated repository managed by the change control software (Subversion). Within each project we create a standard set of forums (bulletin boards) for the student to collaborate on. Our identification and usage of the boards has evolved over the years and includes our experiences prior to utilizing Redmine. The boards are currently named for the course deliverables our students are expected to produce. They are Background Information, Project Management, Statement of Work, Design, Midterm, and Final Deliverables.

A degree of automation has been developed to facilitate the creation of the projects at the start of each semester, the addition and assignment of students to the projects and, at semester's end, the removal of each students 'active status'. This was done by creating our own Redmine plugin.

5. Implementation

In this section, we shall share some best practices and observations from our application of the Redmine package. It is the authors' hope that these will be helpful to those looking to implement their own collaboration systems using Redmine or similar tools.

We introduce the features of Redmine and some best practices, which are presented in the following sections, to the students at the beginning of each semester. It takes approximately 20 minutes and, in general, is sufficient for students to get started. Students, who had difficulty in using Redmine, typically needed the following refresher instructions in basic skills taught in the introductory engineering design course:

- Understanding of maintaining a good design or lab notebook – needed for effective use of Forum and/or Wiki
- Project planning, especially decomposing a large task into a set of smaller tasks and identifying task dependency – needed for effective management of issues (tasks)

5.1 Using a Virtual Machine

We recommend taking advantage of Virtual Machine technology to allow for experimentation prior to a production installation. Using the VirtualBox [6] technology provided by Oracle allows one to create a virtual machine, install the operating system and then take a 'snapshot'. The user may then add more software and configuration data and if that iteration is deemed incorrect, the system can be reverted back to that snapshot and the process repeated as needed until a suitable implementation is found. Moreover, this approach allows us to quickly recover the system in case of a server hardware failure.

We also suggest that Virtual Machines be used when upgrading. To ensure the safety of each upgrade, our approach has been to create a virtual machine with the same version of operating system as our production machine. By then installing the running version of Redmine, a copy of our production database and a small set of uploaded files one can create a mini-clone of the production system. A snapshot of the virtual machine in this state can be used to 'roll back' the changes if needed or to allow the upgrade to be repeated for practice. When testing a new version, one must be sure to verify both Redmine itself as well as all plug-ins currently in use.

5.2 Issues and Project Management

We observed that teams making use of the Redmine Issues functionality are better organized and yield a stronger team output. Based on our project management experience, we coach our students to put in a meaningful set of issues with reasonable start and end dates and to ensure that each issue has an owner. This allows the team to use the automatically generated Gantt Chart to track their project's progress and to make progress reports to the faculty advisors.

By adjusting the configured workflow or by establishing a simple set of conventions, it can be enforced that the owner of an issue is not allowed or able to close it - that either the Project Engineer or a designated team member must confirm that the issue is actually complete before it is closed. This reduces 'issues churn' where minimally useful issues are opened and closed so

that a team 'looks busy'. Validating the issue's actual completeness also provides a level of quality assurance.

5.3 Wikis

Wikis can be used for a variety of academic purposes. We have found that two general uses are most helpful in our program. Because all team members can make edits to wiki pages, it can be useful for collaboratively writing reports. A drawback to this is the challenge of obtaining the desired formatting within the wiki. A plug-in called 'DocPu' [7] is available for document publishing from one or more wiki pages, apply formatting commands and product one of multiple outputs.

The second and most common usage in our program, is for 'how-to' or knowledge base creation. As all team members are able to modify the pages, students are encouraged to add information as they discover it. A wiki page containing development or documentation standards helps the entire team utilize a common approach. While the technology of a project may be familiar to the students, the application domain may not be. A project glossary of terms and abbreviations help the team with the commonly used 'lingo'.

5.4 Forums

The students have been very successful using the forums for both technical discussions and design reviews. Students are able to collaborate with faculty staff and sponsors following the industry model of multiple contributors who access the discussions from multiple locations and at various times.

The forums are represented in Redmine in a typical bulletin board fashion using the 'threads' model. This maintains the integrity of a discussion by interweaving posts from each participant, but it can make it harder to know where to find information. Therefore, students are encouraged to create a cross reference or index of the forum postings in the wiki. This is, unfortunately, a manual process.

5.5 Version Control - Repository and SVN

Subversion [8] was selected as the tool for change management. Although Subversion can be a standalone tool, students can use their Redmine account to access it and monitor activates using the Repository tool. Each project in Redmine has a separate Subversion repository automatically created by Redmine when the project is first defined. Figure 1 shows a sample repository structure that meets the multidisciplinary needs of our projects. The Tags folder would include a copy of the working folder made at the end of each semester as a snapshot of the project as left by each project team.

Maintaining separate repositories allows us to secure and isolate the intellectual property of each sponsor. This also eliminates having one team accidentally or intentionally modifying the repository of another project team.
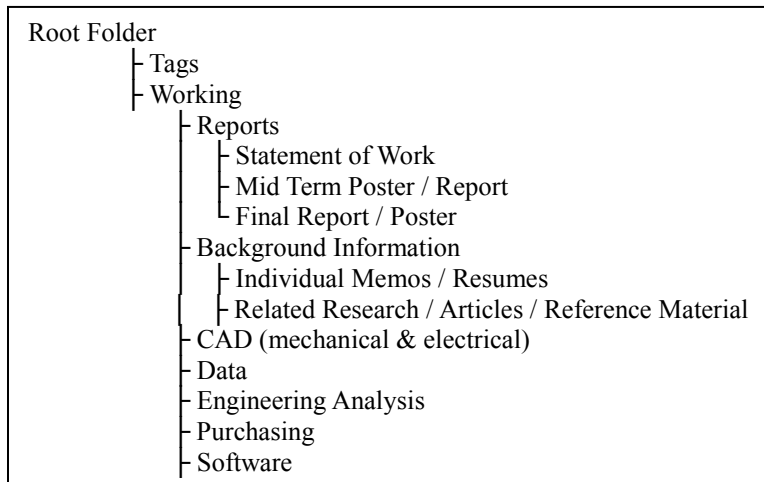
```
Root Folder
        ├ Tags
        ├ Working
                ├ Reports
                        ├ Statement of Work
                        ├ Mid Term Poster / Report
                        └ Final Report / Poster
                ├ Background Information
                │       ├ Individual Memos / Resumes
                │       ├ Related Research / Articles / Reference Material
                ├ CAD (mechanical & electrical)
                ├ Data
                ├ Engineering Analysis
                ├ Purchasing
                ├ Software
```

Figure 1 Sample Subversion Folder Structure

5.6 Inventory Management Support

As part of our continuous improvement efforts, we periodically review the available plugins for Redmine [9]. Prior to the Fall 2011 semester, we added the 'Booty Bay' plugin that allows us to manage our loaner equipment and textbook library. Each record consists of the name and description of an item and the current holder. We also created 'phantom' users bearing descriptive location names, such as 'Storage.Cabinet.2', so that it can be used to indicate the item's location when it is not borrowed.

By replacing our previous paper based sign-out log with 'Booty Bay', students are able to locate any books/equipments they need. Using the recorded user names and email addresses, an academic advisor can quickly collect these items back to the Lab and may facilitate sharing/scheduling books/equipments as needed. These records have proven to be extremely effective as all items borrowed during the Fall 2011 semester were returned by their borrower.

6. Monitoring and Assessing Students and Their Progress

When the system was not available, faculty advisers heavily relied on face-to-face meetings in scheduled classes to assess students' progress and problems. The system allowed faculty advisors to monitor the students' progress made between classes. Therefore, in classes, the students and a faculty advisor were able to avoid trivial housekeeping issues and focus their attention on important design issues.

6.1 Monitoring Student Participation and Progress

One of our initial criteria for selecting Redmine was the ability to monitor and observe student usage, participation and progress. Redmine has been very helpful in accomplishing this through our use of its built-in capabilities and our creation of a supporting plug-in.

Redmine includes the ability to display the last date that a user signed into the system. In a typical semester, students would be expected to use the system every day or two. Students who have not recently accessed the system can be encouraged to increase their online participation, both for a technical contribution and as part of the course communications requirements.

Redmine allows us to review students' activity by either a date, such as Figure 2, or a user, such as Figure 3. The system can display all activity or only specific types, such as wiki edits or issue related updates. Faculty and staff can evaluate what a team was doing or not doing on a particular day in matter of seconds and review details by clicking a specific activity. We point out to the students these features are also available to their team mates, suggesting that activity or inactivity are both visible throughout the team, not just to the project coaches.
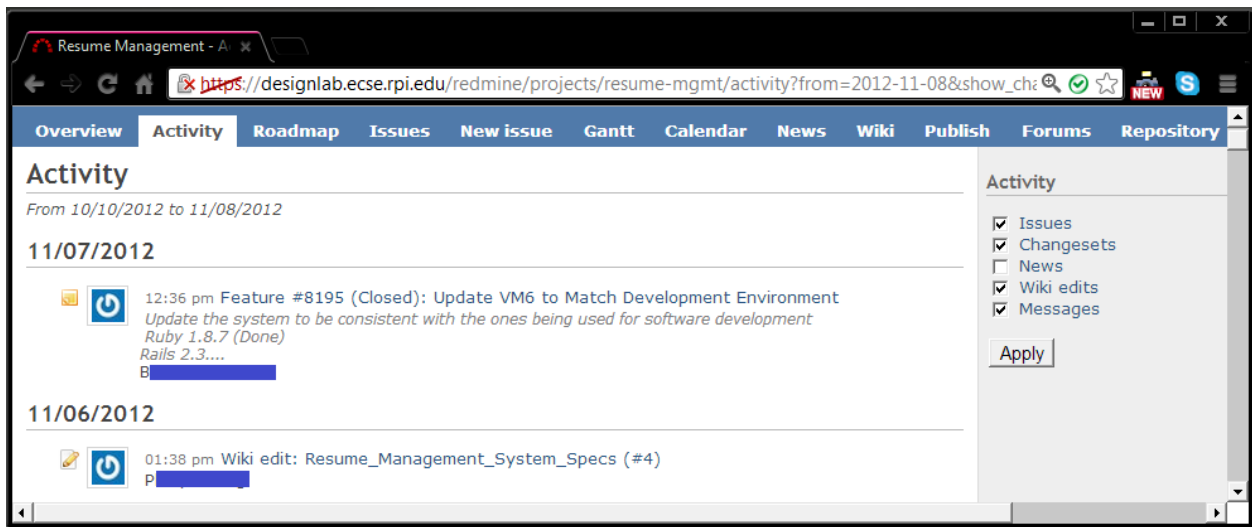

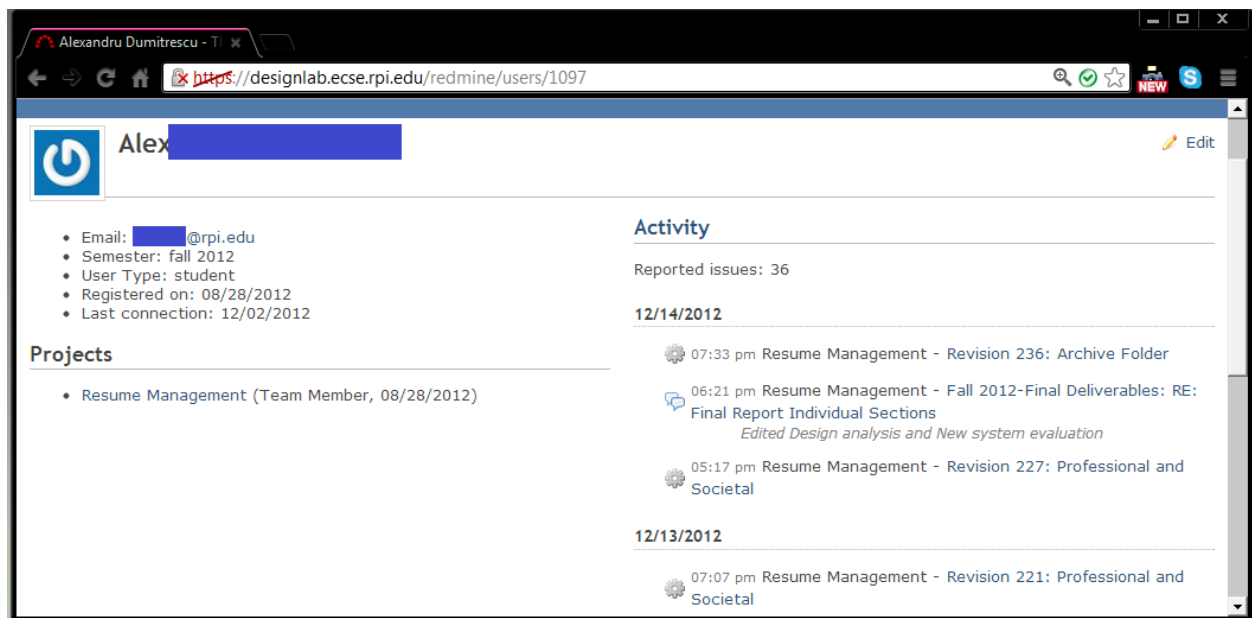**Figure 2 Showing Activity by a Date**

Figure 3 Showing Activity by a User

When evaluating these contributions, we look for timeliness of replies and origination of new material by the student. The system has proved effective for student coaching by reminding students that signs of continuous participation would include their posting of original content as well as thoughtful analysis and feedback on the contributions of others. It is important that monitoring of postings not simply be a numbers game; the content and regularity of postings are more important than sheer volume.

Since all activity is date and time stamped, multiple activities can be combined to a single post. For example, we often repeat an edit and save cycle to revise a wiki page. Figure 4 shows that this student edited the Application_Setup page twice in two minutes on November 12. Hence he revised two wiki pages rather than three wiki pages on November 12.
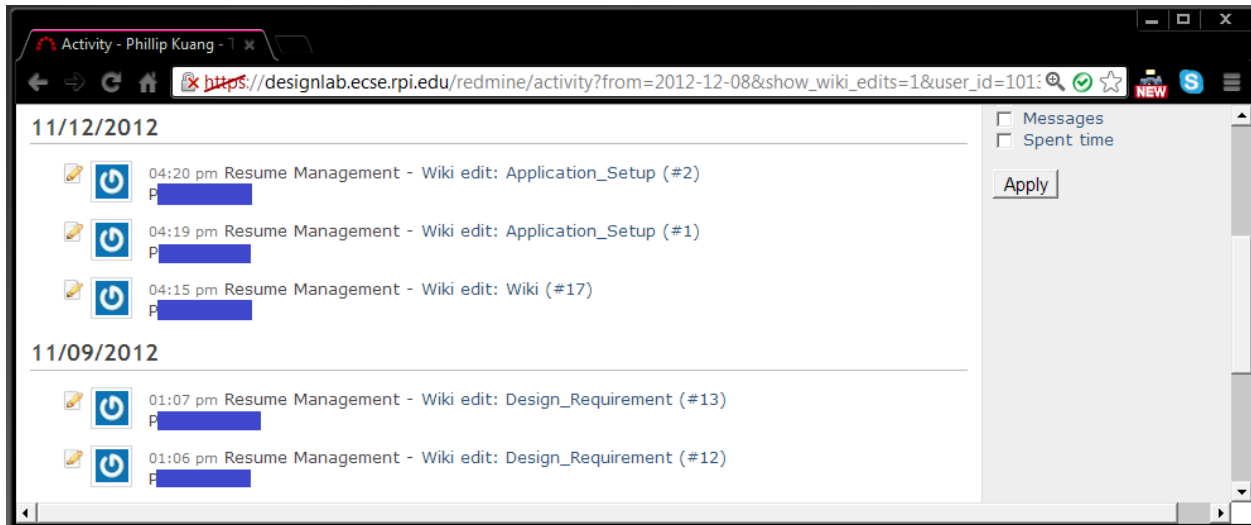

Figure 4 History of Wiki Pages Revisions Made by a Student

Similarly, students can up load each file separately or multiple files at once. Hence, the total number of days a student uploaded files to the repository may be more insightful for students' efforts as shown in Table 1. This four person team was randomly chosen from 19 teams who worked on capstone design projects in the fall of 2012.

Table 1 Total Number of Days Students Utilized the Features During a Semester

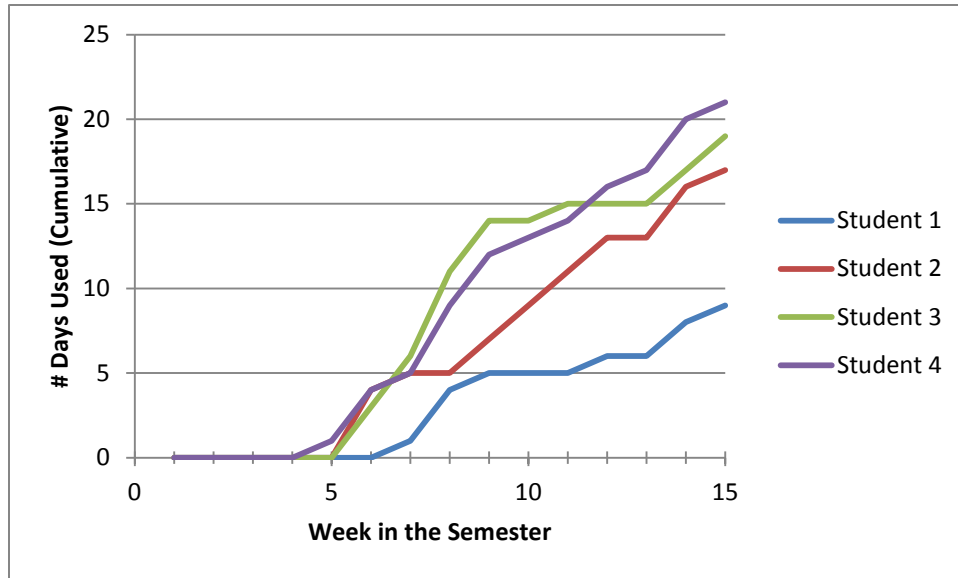| Student	Forum (Messages)		Wiki		Repository (Change set)		Total	
	#Actions	#Days	#Actions	#Days	# Actions	#Days	#Actions	#Days
1	28	7	10	7	14	9	32	23
2	24	14	23	7	51	17	57	38
3	10	8	7	2	33	19	53	29
4	24	14	10	3	47	21	63	38

Figure 5 Total Days Utilized the Repository Over Time

Overall volume of postings is, however, loosely related to a project's overall activity level. However, it does not indicate the quality of work. While a high number posts by a student is not a singular indicator of success, a low number of posts (or inactivity) generally is an indicator of a problem. Those with substantially lower activity levels can be reminded of the importance being actively involved in their project. For example, Figure 5 shows that Student 4 did not upload any design artifact to the Repository for two weeks (in Weeks 10 and 11). This coaching opportunity itself affords the opportunity to have one-on-one involvement which may be more important for some students.

6.2 Monitoring Project Progress

Based on the authors' many years of working with or within industry, the value of using the Redmine Issues and Gantt charting feature has been clear from the start. Information gathered for creating and updating Gantt Charts is automatically gathered as Issues are updated. We have found it necessary to clarify to students that Redmine's terminology is not identical to those traditionally used in project management as seen in Table 2. Students are better able to benefit from the tools once they have a clear understanding of the traditional terms and methods and how these are represented by Redmine.

Table 2 - Traditional Project Management vs. Redmine Terminology

Traditional Terms	Redmine Terms
Tasks	Issues
Milestones	Versions

To ensure that demonstrable progress is being made, we coach our students to include mention of the work completed when the percent complete number is adjusted. Such a note may include a link to a forum posting, a link to a wiki page that has been updated or a link to a revision in Subversion where the issue has been addressed. As part of quality control feedback, the percent

completed may be reduced if a team member or coach does not agree as to the effectiveness of the previous update. This may lead to improved or repeated testing, additional analysis or other appropriate work.

We have built a small number of standard queries to help teams manage their issue lists. Typically, issues not having a responsible owner do not see much progress. As a result, we have created a search for issues without an assigned owner. To help students cut through a long list of issues, another query selects those issues assigned to 'me', thus providing the student with their personal to-do list. Redmine includes two charts related to issues - one to show how the quantity of issues is changing over time and another that indicates how long issues are remaining open. The first graph, such as Figure 6, typically shows a slow rise in the number of issues as the students 'ramp up' in the project. As the students come to understand the problem and generate solution concepts, the number of issues will rise. A plateau may occur as the students become involved in bringing their concepts into practice. When the students have selected a concept to follow, the number of issues will steadily rise again as they record and resolve implementation problems and refinements.

We have found that students and coaches can better track progress when well defined milestones, or 'versions' as they are called in Redmine, are used in the project plan. When new issues are created, they may be tied to a version. Redmine can then list the issues that must be achieved to reach a version, providing a 'to-do' list for the team. Issues may be prioritized and, as needed, reassigned from one version to another. This overall list of the versions and their related issues provides the team with a project road map. Our experience confirms that teams who pay appropriate attention to this project management yield stronger project outcomes.

While the typical industry practice is to use Gantt charts to visualize project schedules, student may be more comfortable with a calendar based view. Redmine makes this alternate view readily
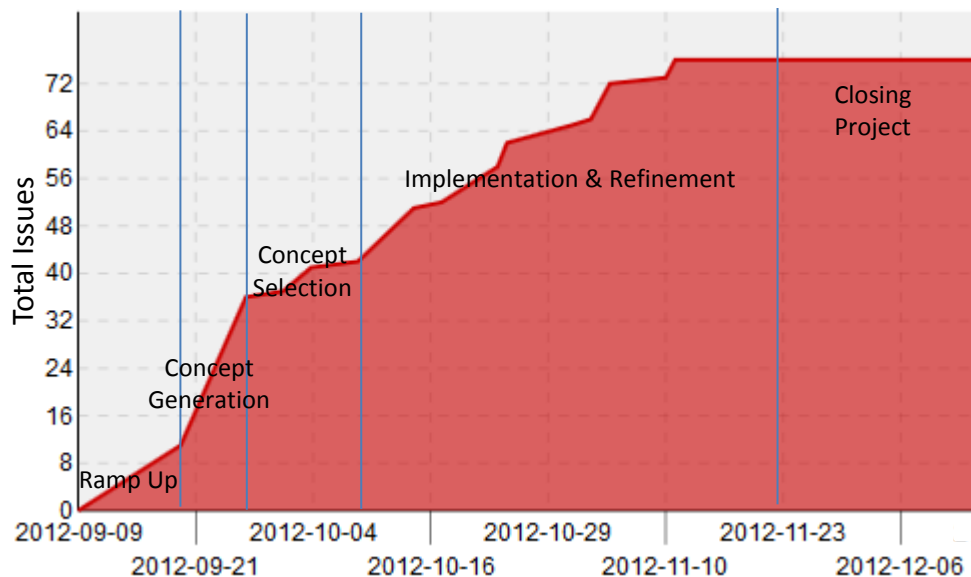

Figure 6 Total Issues Over Time

available. By simply viewing the calendar, students can see which issues are currently active.

Redmine offers an additional variation on standard project management by making a 'Roadmap' available. Here one can see the Versions (milestones) that have been identified for the project as well as a list of the issues needed to accomplish each Version. Issues that have been completed are marked by a line drawn through them much as one would strike out completed items on paper. When a system is divided into subsystems, a version can be used to manage a subsystem development also.

Conclusions

In this paper, we presented best practices in applying an on-line collaboration system to a capstone design course. We also introduced the configuration settings and operational procedures that allowed us to provide a solid academic experience for our students while minimizing the support needed to keep the system operational. Our use of Redmine has proven to be productive and powerful for both students and academic advisors. Academic advisors can continuously monitor students' efforts and progress and the quality of students' design artifacts. It allows academic advisor to provide timely and helpful feedback to students.

Although these students were not widely separated by distance, their class schedules made it difficult to hold full team meetings outside of regularly scheduled class times. They learned to leverage Redmine to work independently but share results across the full team. This experience helps students transition to the business world after graduation. Our students learn the importance of collaboration and gain exposure to the tools that can facilitate this.

References

1. Sheppard S, Macatangay K, Colby A, Sullivan WM. *Educating Engineers: Designing for the Future of the Field*: A Wiley Imprint; 2009.

2. List of collaborative software. *Wikipedia*. January 25, 2012. Available at: http://en.wikipedia.org/wiki/List_of_collaborative_software. Accessed January 28, 2012.

3. Engineering Accreditation Commission. CRITERIA FOR ACCREDITING ENGINEERING. *www.abet.org*. October 29, 2011. http://www.abet.org/uploadedFiles/Accreditation/Accreditation_Process/Accreditation_Documents/Current/eac-criteria-2012-2013.pdf. Accessed January 27, 2012.

4. Kanai J. Web-Based Collaboration Tool in Engineering Education. Paper presented at: Proceeding of ASEE Zone Conference 2008, 2008; West Point, NY.

5. Redmine software home page. *redmine.org*. Available at: http://www.redmine.org/. Accessed February 23, 2012.

6. VirtualBox sfotware home page. *virtualbox.org*. Available at: https://www.virtualbox.org/. Accessed February 23, 2012.

7. DocPu, Document publishing plugin. *Redmine.org*. Available at: http://www.redmine.org/plugins/redmine_doc_pu. Accessed February 23, 2012.

8. Apache Subversion. *Aparche.org*. Available at: http://subversion.apache.org/. Accessed February 23, 2012.

9. Plugins Direcory. *Redmine*. Available at: http://www.redmine.org/plugins. Accessed February 23, 2012.