# *ELIS – Managing Enterprise Level Learning Programs with Moodle*

*Mike Churchward*

*Remote-Learner.net, USA/Canada/UK, mike@remote-learner.net*

## Abstract

Moodle as a learning management system does a superior job managing individual courses, their content and the users within them. Enterprise users frequently have needs beyond the course, especially with managing its users in organizational structures into learning programs.

This paper looks at ELIS, an application add-on for Moodle that provides a way to manage Moodle users and courses into targeted learning programs and report on progress in those programs. It will describe the use cases that led to ELIS, the work done to define and build it, the feedback mechanisms to help define the feature roadmap and the process used to build and manage it.

## Keywords

Moodle, Enterprise, Reporting, Compliance, Certification

## Background

Many of our clients wanted to use Moodle, but found Moodle lacked the tools they wanted to manage large groups of users over the history of their education, and to easily archive and access the educational progress of those users. These clients needed to be able to group courses together to achieve specific larger learning goals. These curricula or learning programs could also have dependencies. Tracking the progress and achievement among these courses is crucial. In corporates and public sectors, this need is frequently part of compliance training and skills certification. Additionally, managing the users in these programs needed to be done in an organizational structure that closely matched the enterprise's structure. And different managerial roles needed to be assigned within these structures.

What became ELIS, originally started out as a project to provide an on-line university with the ability to manage its courses in curricula, and manage its users in the system in clusters. Reporting was also needed to provide transcript-like reports and curriculum progress reports.

Moodle has many third-party plug-ins available. Reviewing all of the known add-ons available, revealed none that could provide curriculum-like management within Moodle. There were commercial student management systems and CRM's that could provide similar functionality. The possibility of developing integration layers between Moodle and those systems was explored, but the costs of the commercial systems and the efforts to maintain any developed integration layers proved prohibitive. Also, there were other LMS systems that had this functionality native or available to be added, but these again were high-priced commercial systems and would not meet the base cost criteria. And since one of the key goals was to keep the solution in the Open-source world, closed, commercial systems were ruled out. As such, a project to develop the functionality as a series of Moodle add-ons was created.

It also became apparent that many of our corporate clients had a need for similar functionality in order to meet their various compliance and certification needs within their organizations or for the organizations they were responsible to monitor (in the case of governing boards). A project was setup to expand the curriculum and cluster system built for the on-line university into ELIS – Enterprise Learning Intelligence System.

The continuous evolution and development of ELIS was and is managed through continuous "in-production" research using key "anchor" clients, who represent the broadest coverage of Moodle with ELIS use cases.

## ELIS General Concepts and Structures

Addressing the basic requirements defined above, we organized the basic structure of ELIS into six essential structures:

- Course Description (previously, “Course”)
- Class Instance (previously, “Class”)
- Program (previously, “Curriculum”)
- Track
- Learning Objective (previously, “Completion Element”)
- User Set (previously, “Cluster”)

In earlier versions of ELIS, the “previous” terms noted above in brackets were used. This document uses the newer terms as much as possible, although the older terms may appear in some places, particularly imagery.

The following diagram shows the relationships of course descriptions, class instances, programs and tracks:

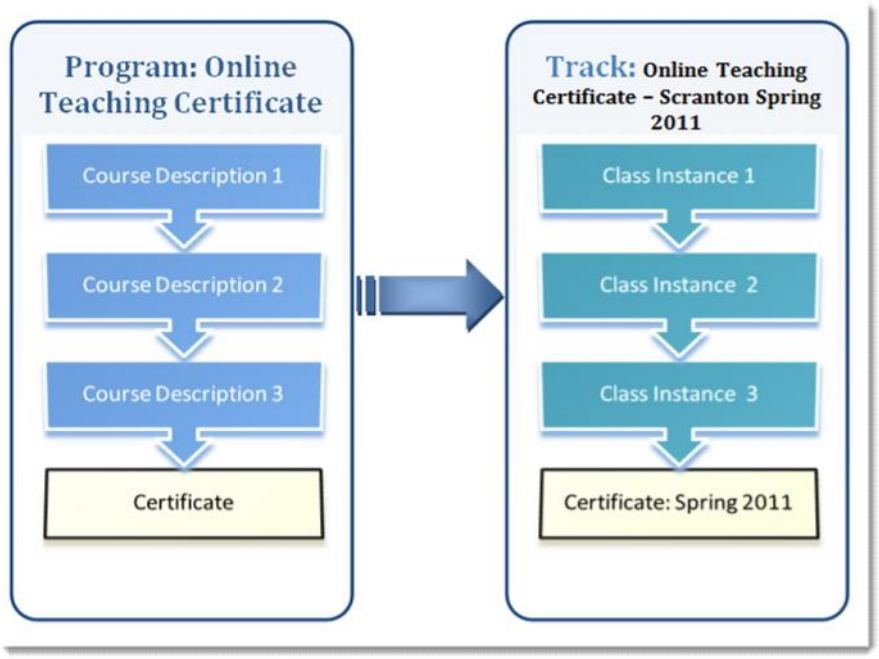


**Figure 1: Learning Program Relationships**

### Course Descriptions

Course descriptions define the meta-data, credits, duration, and learning objectives for a course of study. This is a software implementation of the course descriptions that are frequently published in course catalogues or listings of required compliance, skill, or competency courses. Users are not enrolled in course descriptions. They are enrolled in class instances, which are created from course descriptions.

Standard Moodle forms were developed to facilitate the entry of all course description information such as name, code, description, number of credits and required grade. Course descriptions can be linked to a Moodle template course for automatic creation of Moodle courses when class instances are created from the course description.

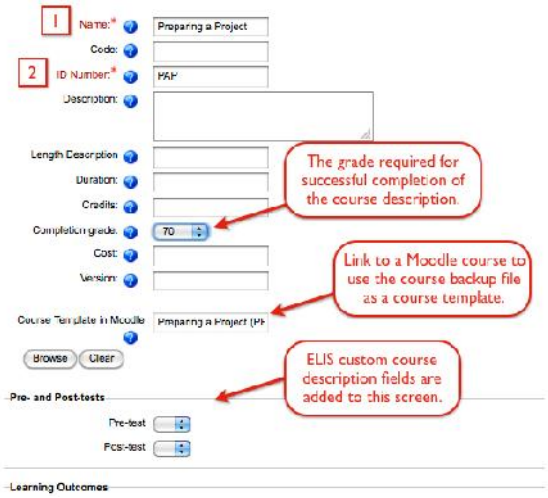


**Figure 2: Course Description Data**

## Class Instances

Class instances are instances of course descriptions. Students are enrolled, and results are recorded and stored in these. Class instances can be associated or connected with Moodle courses, or can be a record keeping and reporting tool for face-to-face courses which have no online component. A class instance can also have elements from a Moodle course and have elements recorded manually from face-to-face meetings or other external events, providing support for blended learning.

Standard forms were created to facilitate the manual creation of class instances from their course templates. The additional information, such as start and end date and maximum number of students can be entered at the same form. Once created, extra data is then available to be added and edited on the class instance. This includes connecting the class instance to an existing Moodle course, by selecting it from a drop-down. This connection, facilitates the automatic data exchange from ELIS to Moodle and vice-versa.

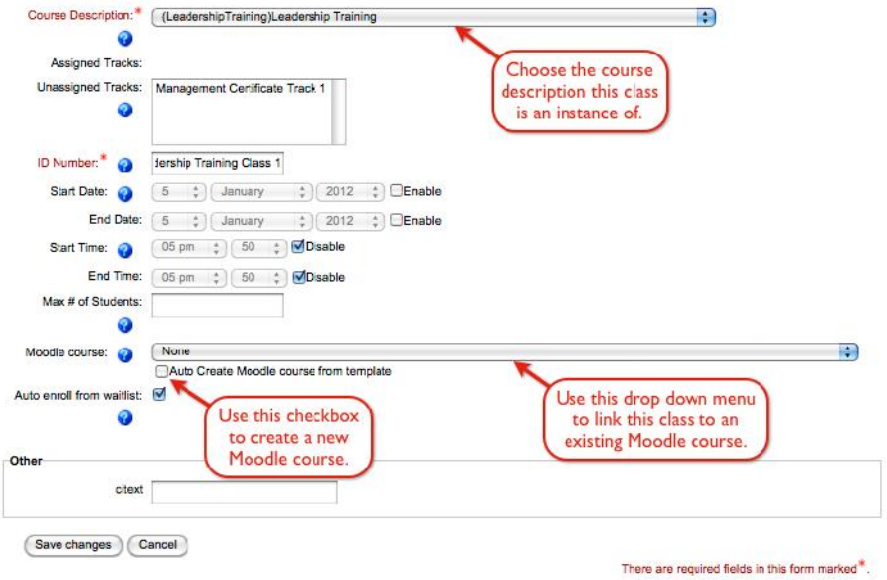


**Figure 3: Class Instance Data**

Enrolments can also be managed. An enrolment record consists of a user and their progress information – completion time, completion status, grade, credits and the status of any learning objectives. While a class

instance is connected to a Moodle course, users can be automatically enrolled into the Moodle course from ELIS, or in a Moodle-centric setup, users can be enrolled in the ELIS class from the Moodle course. Any connected grade information, such as Moodle grade elements with a matching id code to an ELIS learning objective, are automatically synchronized. The key to ELIS is that this class instance information can be kept independently of the future existence of the connected Moodle course.

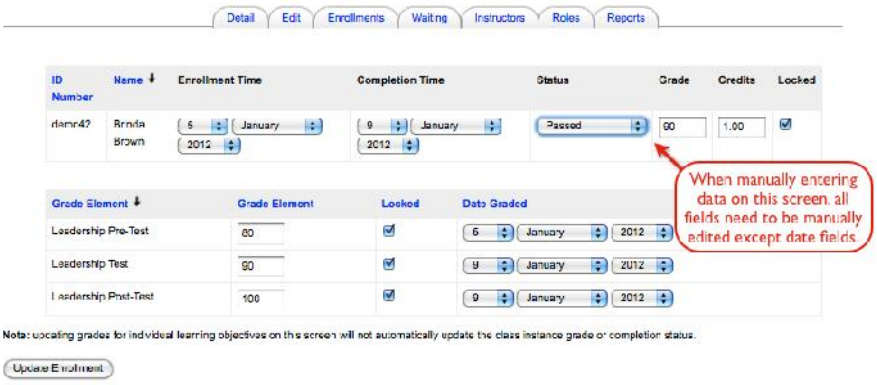


**Figure 4: Class Enrolment and Learning Objective Data**

## Programs

A program in ELIS is a series or group of course descriptions. They are a set of linked courses used to track specific learning goals. For instance, a student might have to complete a series of courses to get a certificate, to demonstrate their competency for a job, or to maintain their certification on a critical skill. Programs are used to define these series, independently of date or a particular set of users.

Forms were created to allow for creation of programs. At its basic level, a program has a name, id code, description and the required credits for completion. It also contains one or more course descriptions, each defining whether it is required for completion, the position to display in and frequency information. Programs allow for reports to track progress of the users assigned to them.

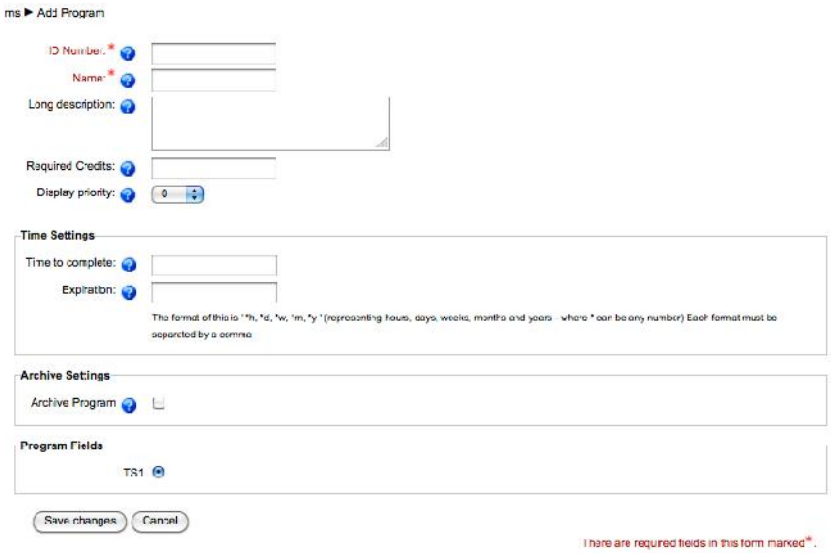


**Figure 5: Program Data**

## Tracks

A track is a given instance of a program - a set of class instances that are given a particular schedule, with a particular set of users. Tracks are used to manage enrollments, start and end times, reports, etc. Wherever it is required that a group of users take a particular program of study, a track is quick way to create classes and manage enrollments for that program.

One of the strong functions of tracks are their use in automation. When a track is created from a program, the program has all of the course descriptions defined. Each of those course descriptions can have Moodle course templates assigned to them. When the track is created, a “create all classes” option is available. When this is used, all of the ELIS classes can be created for that track automatically. Additionally, any Moodle courses can be created from course descriptions with defined Moodle course templates, and connected to the ELIS classes. Further, any users assigned to the program, or to the track, can then be automatically enrolled in the ELIS classes and the Moodle classes.

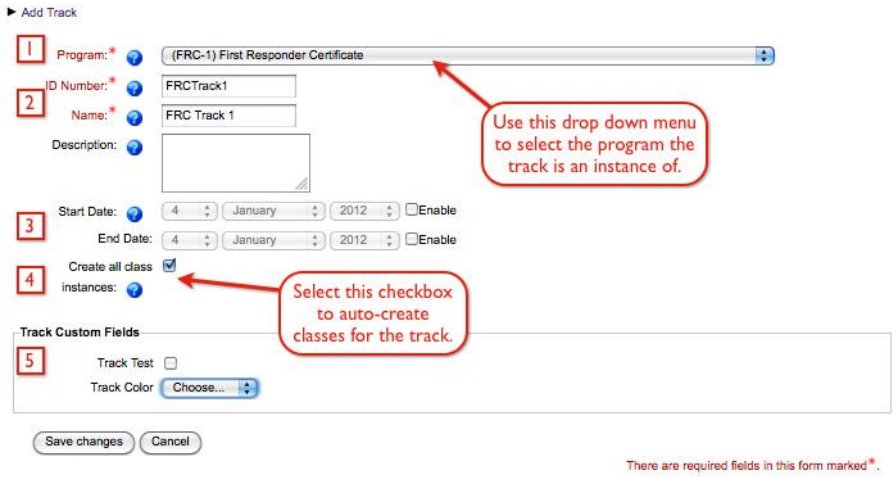


**Figure 6: Track Data**

## Learning Objectives

Learning objectives operate at the ELIS course description level as a method for assessing learner competencies, the key concepts or ideas that learners should take away from a course. They exist outside and above the Moodle course content level, at the ELIS course description level. Learning objectives can define required activity in a course needed for a student to be considered complete. How those objectives are met can be defined in the class instances themselves. For example, at the Moodle level one instructor might decide that a particular learning objective for the course would be met by a quiz, while another might decide it is met by a series of assignments. Learning objectives become part of all class instances, and can be manually managed with a class instance enrolment record. This facilitates recording of non-Moodle classes as well.

Learning objectives can be defined for each course description using a Moodle form. If a course description has a defined Moodle course template, then the learning objective can be easily linked to a Moodle grade item using the browse feature to look at that Moodle course’s grade items.

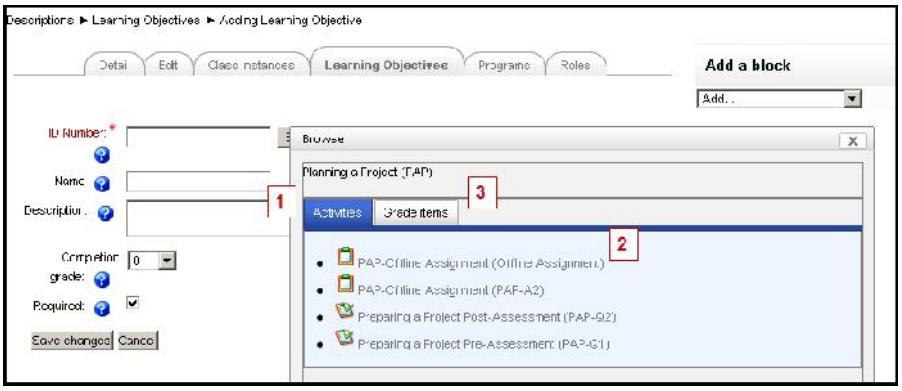


**Figure 7: Learning Objective Data**

The learning objective data is permanently stored with a user’s enrolment record in ELIS, even if the associated Moodle course is removed later.

### User Sets

User sets are flexible site level groupings of users. They can be used to set up hierarchies of users, such as the organizational structure of an enterprise organization. These hierarchies can be role and permission based, allowing specific users to be assigned reporting, managing and other administrative duties over selected sets of users by defining roles at appropriate context levels. User sets also facilitate the management of groups of users through programs, providing easy methods to enrol groups into appropriate courses and programs.

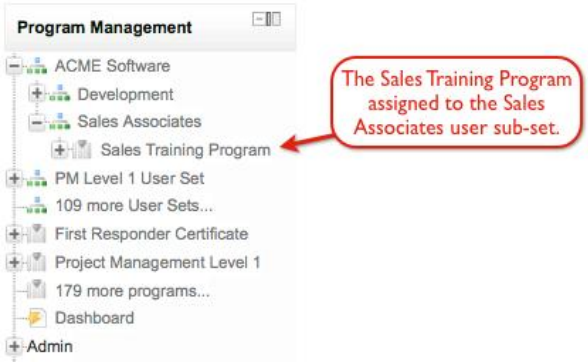


**Figure 8: User Set Hierarchy**

User sets can be used to automate movement of users through the learning program by assigning to programs, tracks and classes. Users can be automatically assigned to user sets using profile data settings. And, user sets can be automatically associated with Moodle groups and groupings in Moodle courses. The data for user sets can be managed using Moodle forms.

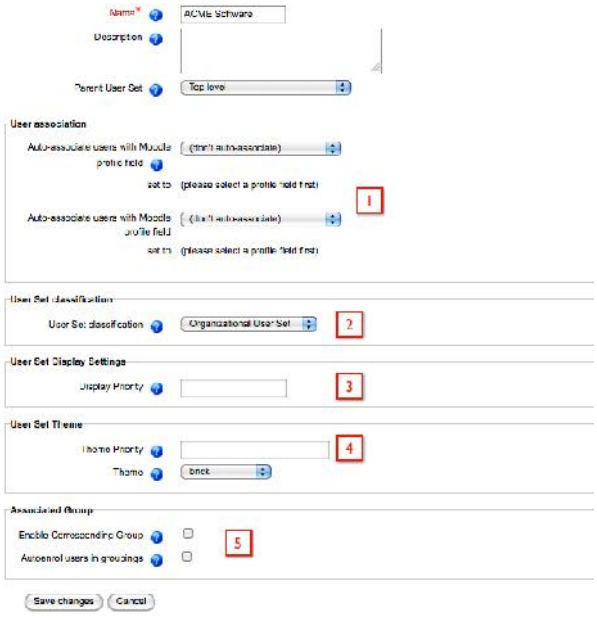


**Figure 9: User Set Data**

## ELIS Evolution

Moodle is key to ELIS. We chose to develop ELIS using Moodle's API's and structures for several reasons:

- We were already "experts" in the platform, so no learning curve.
- Moodle is the central component to delivering the learning of the system.
- As Moodle evolves, functionality that ELIS provides is occasionally adopted into Moodle, allowing ELIS to migrate that function away.
- Releasing ELIS as open-source, provided the community as potential enhancers.

There were and are problems to deal with by using Moodle's API's:

- Changes "under the hood" to Moodle, that don't affect users, can greatly affect development (2.2 capabilities model).
- ELIS development lagging behind Moodle releases.
- Management of multiple releases.

### ELIS Generation One

The early releases of ELIS focused on learning programs and reports. Structurally, ELIS code was built in a separate directory below the Moodle home directory, and managed with a block plug-in. Minimal core changes were done in order to keep the code management process as simplified as possible.

The initial release had all of the previously defined structures with basic functionality built around them. The first releases of ELIS, required much of the data to be entered manually. The focus was on getting a working learning program set up, and then allowing the reporting to do what was necessary. Users were manually entered into user sets, classes were entered into tracks, connections were made to Moodle courses and learning objectives were connected – all from the UI.

For this generation, ELIS was integrated with an external business intelligence and reporting engine, Jasper. Jasper had an open-source, Java based application that provided interfaces to other applications. A number of "canned" reports were created to distribute with ELIS, that would report on the most commonly asked for data.

### ELIS Generation Two

After several months of piloting the first releases of ELIS, the "anchor" clients were interviewed to learn their experiences, and determine what needed to change. The three biggest issues were the complexity of setting up the system, the amount of manual labour to integrate with large amounts of enterprise data and the difficulty configuring the reports to their specific needs.

To address setup complexity issues, we added training programs, user manuals, integration applications and more automation. The integration and automation additions also helped to solve the manual labour issues with large amounts of data. For reporting issues, a new reporting engine was built and released.

*Automation*

One of the largest improvements to ELIS was adding more management tools around user sets. Typically, in a large organization, users were managed into learning paths in defined groups. User sets made sense to do the same. And, in these organizations, there was usually data that defined what user sets users should be in.

ELIS already had the ability to move users into programs and classes by assigning them to user sets, and then assigning those user sets to tracks and classes. We created automatic ways to add users to user sets through the manipulation of user profile data. Essentially a user set could be configured to add all users who had certain profile data fields set to specific data. When a user's profile data became that value, they were automatically assigned to the user set. If that user set was already assigned to tracks or classes, then the user would be enrolled in those classes. This allowed user sets and learning programs to be managed using profile data settings.

Once this function existed, it then became easier to create new methods of managing profile data, so that the process could become more automatic. Using integration points, enterprise data could be imported into ELIS/Moodle through a variety of conduits (web services, text upload). If this data contained profile data, changing the data allowed for automated movement through the system.

Other automation that was added was the auto-creation of ELIS and Moodle classes through the track elements. By linking course descriptions to specific Moodle course "templates", when class instances are created from those course descriptions, ELIS can use Moodle's backup and restore functions to create new Moodle courses and ELIS classes and link them together. When tracks are created from programs, this can automatically create whole new series of classes and enrolments using this method.

*Reporting*

Although Jasper provided a flexible business intelligence engine, allowing users to create and manage their own reports, it had a very steep learning curve. It also wasn't easy to theme, making it difficult to create a unique looking report for each client. It became apparent that most clients simply wanted ready-made reports that looked like their sites.

While not abandoning the Jasper integration, we set it aside to build a flexible PHP-based report engine that would allow us to build reports easily to clients' requests. This required programming, but most of our clients didn't want to build their own reports – they simply wanted them to be available easily.

### ELIS Generation Three

The latest incarnations of ELIS, along with being built on the Moodle 2 technology, contain even more automation and more integration options.

We have built a results engine into ELIS, that allows for learning paths to be defined adaptively, based on progress through other classes and learning objectives. The results engine lets you define a goal based on an ELIS learning objective or class grade. When the goal is achieved, actions can be defined to enrol the user in a specified track, enrol the user in a specified class and/or set a specific profile data field to a specific value. This means that learning programs can be set up to drive users into other programs or tracks based on their results in a course or activity. And data can be set to use in reports or other possibilities.

The next generation of ELIS integration point has been developed to allow import and export from more external systems in a pluggable way. This system allows for user, progress, enrolment and other data to be easily exchanged with ELIS and Moodle. The system is pluggable, meaning that conduits can be defined and developed for multiple external systems using multiple data methods.

### ELIS Development, Management and Distribution Process

One of the biggest problems managing the development of ELIS, is maintaining it to the current versions of Moodle. While Moodle strives to be supportive of the external developers as well as its users, out of necessity its users come first. This means that occasionally, a release changes something fundamental under the hood that impacts the way ELIS was written, even though it does not visibly affect the user experience (except maybe to improve performance. We also have to be careful of any core changes to Moodle made to support some of the features of ELIS. And there is simply integrating in the latest maintenance releases of Moodle.

To do this, we have adopted similar processes to Moodle HQ. We maintain our own git-based repositories that draw from the main Moodle repositories as well as our own. We also export just the ELIS portions of the code to a community repository located on Github. The repository setup has allowed us to create a release and distribution process that automates (as much as possible) the integration of all of the code bases.

We have spent a great deal of time creating unit tests in all of the ELIS code. We have also created a large library of user tests, based on use cases, for all of the ELIS functionality. These tests allow us to more easily discover problems when integrating in new Moodle releases. These tests have allowed us to cut down our testing time from months a couple of years ago to less than a week now. We are also in the process of automating our user tests using Selenium-based automated testing.

Lastly, we have based our entire development process on Agile/Scrum processes. Like Moodle, we use JIRA and its agile tools to manage our development releases into sprints. This process has allowed us to maintain schedule by breaking features down into smaller, easier projects. This process has reduced the stress on the presonnel, allowing them to see frequent successes rather than constant delays and failures.

### ELIS Future

Our goal is to add more learning program automation, and more adaptive learning features. We are designing more dashboard style interfaces, ideally that fit into the My Moodle interface, that will help users and administrators manage their way through their programs. And, we want the dashboard to have more on demand reporting.

## References

Remote-Learner.net (2011) ELIS 2.0 Manual.
http://rlcommunity.remote-learner.net/mod/book/view.php?id=63 [viewed May 2012].
Remote-Learner.net (2011). Anchor Client Research.
http://wiki.remote-learner.net/display/ELIS/Anchor+Client+Research [viewed May 2012].
Remote-Learner.net (2012). ELIS Roadmap 2012.
http://wiki.remote-learner.net/display/ELIS/ELIS+Roadmap+2012 [viewed May 2012].