Providing “Real-World” Software Engineering Experience in an Academic Setting

Deepti Suri
Software Engineering Program
Department of Electrical Engineering and Computer Science, Milwaukee School of Engineering
1025 N. Broadway, Milwaukee, WI 53202
suri@msoe.edu

Abstract – At Milwaukee School of Engineering, the Software Development Laboratory was created with a vision of providing “real-life” software development experience to software engineering student teams. It provides students an opportunity to unite theory and practice while working on large scale ongoing projects in the context of a standardized development process. This paper provides details on how the lab is currently structured, its pedagogical philosophy, its challenges, its rewards and how it has evolved since its inception.

Index Terms – software development, software process and practice, real-world experience

INTRODUCTION

Academicians in the Software Engineering (SE) community are continuing to struggle to make the education they provide to undergraduate students practical and relevant. The debate between the interplay of “theory” and “practice” is neither new nor over. The pressures from the industry to prepare future technology professionals with strong technical background, excellent communication skills, awareness of business and global issues has to be balanced with the academic pressures to increase retention and reduce the number of credit hours without compromising on the quality of instruction being provided. Each of these requirements is challenging in itself. In many cases, these pressures are also conflicting which makes the search for a solution even more difficult.

SE educators have been working hard to provide opportunities in an academic setting where the students can apply SE practice (what to do) and process (how to do it) to realistic development efforts in an attempt to close the gap between theory and practice. There is a general consensus in the SE education community that to accomplish this, students need project experience that extends for more than one academic term in an individual course [9]. Several different approaches are being tried, discussed and debated. Reverse engineering a large existing system, rather than forward engineering a new one has also been considered as an alternative and discussed [2].

At Milwaukee School of Engineering (MSOE), an attempt is made to provide that “real-world” experience in a three-quarter (one academic year) software development laboratory (SDL). The SDL was inspired by Moore’s Real World Lab at the Georgia Institute of Technology [6] [7]. It is designed to emulate world-class software development organizations modeled after the Software Engineering Institute’s Capability Maturity Model [1]. The SDL sequence allows students to take part in ongoing software development projects, for internal or external clients. The students in teams of 3-5 work independently on these client projects while the instructors act as coaches and managers. In addition to being on a development project team, each student is also assigned to a staff team that works on improving the lab's process.

In this paper, we describe our experiences, infrastructure, challenges, evolution etc in the operation of the SDL since its inception. It is meant to provide all the details so that this experience can be replicated at any other institution. This paper is also meant to start a debate within the SE education community on the “best practices” to provide the students with a “real-world” experience.

PEDAGOGICAL PHILOSOPHY

As mentioned earlier, the SDL at MSOE was inspired by and modeled after Moore’s Real World Lab. The original vision for the SDL was to create a software development environment with established processes and procedures, where students could apply skills (both process and practice) from earlier classes to produce software in a systematic and repeatable manner, delivering products on time, within budget and of appropriate quality. Since its inception, the “core” vision of the SDL has not changed though the processes used to implement the vision have incrementally evolved.

There are many components that come into play in SE practice – requirements analysis, software architecture, software design, implementation, verification and validation to new a few [8]. SE process, on the other hand, includes components like planning, team management, team integration, quality management, continuous improvement etc. Each of these components is relevant and important. The students at MSOE learn about each of these components in various classes across the curriculum. However, in each of these cases they are learning some new/different paradigms and usually do not have an opportunity to work on realistic artifacts while using those paradigms. In other words, the students in each of their classes are usually working on scaled down versions of real projects and typically exploit only one
particular phase of the development cycle. SDL provides them with an ideal opportunity to bring all that they learned in other classes together.

As software continues to become a major portion of products and its size continues to grow, professionals are paying close attention to improving productivity and quality. Productivity and defect density levels (total number of defects per thousand lines of code) that are considered good today will most likely be inadequate to keep with this future growth. The industry is striving to improve existing processes and develop new ones which are relevant in the current environment. This makes the interplay of process and practice very important in the SDL.

Humphrey defines software process [3] as “the set of activities, methods and practices that are used in the production and evolution of software.” The concept of software process is not so much concerned with particular tools or methodology; instead, the emphasis is on well defined and controlled processes that can be supported by appropriate methods and tools [5].

In the SDL, students have to work on development of large scale real systems while balancing the interplay of process and practice to deliver good quality projects on time. There is a standardized software development process that the students are supposed to follow in the SDL. The process is based heavily on the Team Software Process (TSP)SM as described by Humphrey [4]. It is important to note that in the SDL students not only learn but also experience (i) the benefits of following the process along with (ii) the dangers of cutting corners. This is a tough balancing act because if not managed properly process can be perceived as bureaucratic overhead that slows down progress without substantial rewards. On the other hand progress can be made on the project but at a pace that neither allows the current cohort of students enrolled in the SDL to see the benefits of their work nor is able to sustain the interest of the stakeholders.

CURRICULAR CONTEXT

The undergraduate software engineering program at MSOE began operation in 1999 and had its first graduating class in spring 2002. The SE program was accredited by the Engineering Accreditation Commission (EAC) of the Accreditation Board for Engineering and Technology (ABET) in September 2002 as one of the first accredited SE programs in the United States. From the beginning, both software engineering process and practice were significant components of the curriculum. The current version of the curriculum is also available at our web site at http://www.msoe.edu/eecs/cese/courses/curriculum.php?progcode=S E. More detailed information on the MSOE SE program is also available at www.msoe.edu/eecs/se/.

The academic schedule at MSOE is based on a quarter system with three quarters in an academic year. Each quarter involves ten weeks of instruction with the eleventh week devoted to final exams. Typical software engineering courses are three or four credits, and most have an associated laboratory session.

The software development laboratory course sequence begins in the winter quarter of the junior year and extends through the fall quarter of the senior year. Upon entry into the SDL, students have already completed courses in programming, data structures, algorithms, design patterns, embedded systems software, requirements, software testing and personal software process (PSP). During their time in the lab, students simultaneously take courses in software architecture and formal methods. After completing the software development lab sequence, students also work in teams on a two-quarter capstone senior design project, which may be multidisciplinary in nature.

Even though the SDL sequence begins in the winter quarter for junior students, the preparations for it typically start much earlier in the fall quarter. The outgoing seniors typically meet with the junior students two-three times during the fall quarter in order to plan for their transition into the SDL. Some of the topics that are covered in these presentations are (i) role of the SDL in the curriculum, (ii) differences between SDL and the other classes they have taken so far, (iii) tools/processes currently being used in the lab and, (iv) current status of various development projects being worked on in the lab and their related technologies. After these presentations, the juniors complete a survey of their skill sets and preferences. Instructors use this information to form new teams. Team assignments take into account individual preferences, skills and attitudes in an attempt to form diverse and balanced teams.

FACILITIES

The physical layout of the SDL includes partitioned workspaces, each large enough for an entire development team of 4-5 students to meet and work comfortably. Each student team has access to several white-boards in their work area. A conference room is used for client presentations, training activities, and laboratory meetings. All students have secure access to the laboratory round the clock. The current SDL facility consists of:

- Team work area #1 (1230 square feet). Work areas for four (4) student teams of 5-6 members, with a display area for materials related to lab projects.
- Team work area #2 (1760 square feet). Work areas for four (4) student teams, instructor workstation, lab servers and related communication equipment, and small-group work areas.
- Conference room and training facility (600 square feet). Conference room with LCD projector and seating for 14-20 persons, for team presentations and client meetings. Convertible to a training facility with a capacity of 21-26 persons.

In addition to the personal notebook computers that all MSOE students have, each team also has desktop computers and access to centralized laboratory servers. Additional server systems are also available for use by development teams if needed. The section on “Infrastructure Support and Tools”
provides additional details on the applications that are hosted on these servers and the students access to them.

A representative work area for the student team and the conference room are shown in Figures 1 and 2 respectively.

The software laboratory facilities are supported in part by a major grant from Johnson Controls, Inc.

LAB EXECUTION

As might be expected, the process used in the lab has evolved over time and will most certainly continue to do so. The way the SDL is administered today is not the same as when it was started. This section provides an overview of the current processes in the lab and an insight into the changes that have been made along with the reasons for making them.

An incremental software development model is used in the lab. Each ten week quarter usually consists of two development cycles of five weeks each in a quarter; except for the first quarter in the SDL which now has only one five week development cycle. As student teams gain experience, they choose an appropriate length for each cycle. Table 1 outlines major activities by quarter, for each student cohort.

The SDL typically enrolls 5-6 development teams with 3-5 members in each. Team members are assigned roles adapted from those suggested by Humphrey[4]: team leader, development manager, planning manager, support manager, contact manager, requirements manager and quality/process manager. These role definitions are helpful, especially for new students, in establishing expectations and providing guidance in carrying out the team’s work.

To assure continuous improvement of lab processes, a structured way to define, improve and evolve the lab processes is needed. For this reason, students are given “staff” assignments in addition to their development responsibilities. For a typical student, 30% of their total SDL time is earmarked for staff activities. Some of the staff teams that are currently in existence in the lab are – SEPG (Software Engineering Process Group), SQA (Software Quality Assurance), TD (Training), PT (Planning and Tracking), ADMIN (Administrative) and ORCH (Outreach).

<table>
<thead>
<tr>
<th>Quarter</th>
<th>First Cycle</th>
<th>Second Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDL-I (winter)</td>
<td>Introduction to lab processes/tools.</td>
<td>Read transition documents¹.</td>
</tr>
<tr>
<td>(winter quarter)</td>
<td>Introduction to basic TSP.</td>
<td>Evaluate the status of the project.</td>
</tr>
<tr>
<td>(junior year)</td>
<td>Introduction to CMMI.</td>
<td>Develop a cycle plan.</td>
</tr>
<tr>
<td></td>
<td>Introduction to post-mortem analysis (Cycle report).</td>
<td>Periodic updates on the execution of the plan.</td>
</tr>
<tr>
<td></td>
<td>Initiate contact with the project stakeholders.</td>
<td>Introduction to team roles.</td>
</tr>
<tr>
<td></td>
<td>Read transition documents¹.</td>
<td>Introduction to a staff group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Research on one KPA for the CMMI report.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Work on product deliverables.</td>
</tr>
<tr>
<td>SDL-II (spring)</td>
<td>Getting better with planning.</td>
<td>Transition preparation for the next team.</td>
</tr>
<tr>
<td>(spring quarter)</td>
<td>Focus on processes that work for the project.</td>
<td>Developing a plan for the new team.</td>
</tr>
<tr>
<td>(junior year)</td>
<td>Incremental deliverables.</td>
<td>Deploying the project.</td>
</tr>
<tr>
<td></td>
<td>Staff activities</td>
<td></td>
</tr>
<tr>
<td>SDL-III (fall)</td>
<td>Wrapping up major deliverables.</td>
<td></td>
</tr>
<tr>
<td>(fall quarter)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(senior year)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The instructor meets with each team for a weekly half-hour status update. Prior to each meeting, the team prepares a summary of information like current milestones, actual effort vs. estimate for each team member during the week, earned value of the project so far, milestones to be completed in the next 1-2 weeks, planned times for the coming week and any questions that the team may have. This weekly status update allows the instructor to evaluate team progress on a periodic basis.

¹ The transition documents also include a tentative plan for the first cycle created by the previous team.
basis and to intervene immediately if there is an indication of some problem. A weekly lab-wide meeting of about fifteen minutes is also scheduled to address any common concerns. The agenda for the meeting is typically emailed to the students about twenty-four hours in advance.

**PROCESS IMPROVEMENT**

Throughout their time in the lab, all students are asked to reflect on their experiences in the lab, identify problems and suggest improvements. Each student must submit at least one process improvement proposal (PIP) each quarter. These proposals have played a very important role in lab’s process improvements. Some of the process improvements that have been made based on student feedback are:

- **Improved planning and tracking tools** – Initially, planning and tracking was done using a spreadsheet tool. After encountering significant difficulties with its usage most teams abandoned its use. A preliminary automated version of the planning and tracking tool was created. It was well received in the lab but was not written for a long shelf life. This precipitated work on a tool called “Leia” (developed in-house) which is currently used in the lab.

- **First quarter format** – For a long time, each quarter in the SDL, including the first quarter, had two development cycles. The instructors recognized that the students would not be able to make significant progress in the first cycle on their development projects as they were settling in the lab and getting familiar with their projects. Based on student feedback, the format of the lab was changed in winter 2006-2007 and the first quarter in the lab has only development cycle. The first couple of weeks are spent on getting familiar with (in interactive sessions) on various tools and processes used in the lab. This change has been well received by the students.

- **Second quarter format** – The second quarter format has evolved from one nine-week cycle to two four-week cycles with one “midterm” week for training and planning and the final week for wrap-up and postmortem activities. One cycle per quarter was not well suited to re-planning in response to changes. Four week cycles, for most student teams did not seem enough to develop, deliver and deploy an incremental deliverable. Currently, each cycle is five weeks long but teams have the flexibility to extend/reduce the length of the cycle based on their needs after consultation with their instructor.

- **Additional instructor help** – The winter quarter of 2005-2006 was the first time that the SDL had six development teams. Prior to that, one instructor used to manage as many as five teams. This arrangement did not allow the instructor to spend adequate time with each team. There are now two instructors in the lab and each instructor has prime responsibility for three teams providing reasonable amount of interaction between student teams and their instructor.

- **Increased and improved communication with the stakeholders** – The stakeholders of every development project are requested to interact with their teams on a weekly basis, with at least one face-to-face meeting every two weeks. This is now a criterion based on which projects are selected and continued in the SDL. In the past, several stakeholders were not scheduling regular meetings with the student teams which were having a negative impact on their morale and productivity.

- **More structured weekly deliverables** – Instructors are paying much closer attention, during the planning phase, to the fact that each team member has structured weekly deliverables. Attempts are also being made to balance student workloads with other major courses in the same quarter.

- **Simplifying web page updates** – Processes have been put in place so that the documents that are checked into CVS are automatically exported to the web. This ensures that updates to documents from the project’s web site become available without manual intervention.

**MEASURING STUDENT ACHIEVEMENT**

Once a team starts executing the plan it has made, it is important to assess how things are progressing. This becomes especially important in an academic setting where student assessment is an essential activity. Currently, the student achievement in the SDL is being measured in multiple ways.

Twice during the quarter, students make portfolio submissions to provide evidence that they are achieving the course outcomes. These submissions are graded on a scale of “Not yet Satisfied”, “Minimally Satisfied”, “Satisfied” and “More than Satisfied” for each outcome and returned to the students. The students are expected to achieve at least “Minimally Satisfied” for each outcome by the end of the quarter. To ensure that students receive feedback in a timely manner, portfolio submissions are made in alternate weeks by each half of the student groups. All students are provided an option of making a final submission in week 9, if needed.

As mentioned earlier, each team meets with their instructors for a weekly review. The instructor monitors the team progress through these weekly meetings. To monitor team function, members complete peer evaluations twice each quarter. At the end of each cycle, teams prepare a cycle report, which is a post-mortem analysis of their performance for the cycle. The students are asked to do some critical self evaluation of their team, in the areas of planning, process and quality. They are expected to analyze what worked well for them in the cycle and what did not. Each team must formulate some concrete suggestions on how to avoid recurrence of the mistakes that they made during the cycle.

For their staff activities, each SDL member delivers a staff activity report each quarter which outlines their contributions towards their staff deliverables on their staff teams. Some of the staff activities in the past have included – creating a risk management proposal for the lab, conducting an audit of transition documents, planning a full-filled gaming night for all SE students etc.
INFRASTRUCTURE SUPPORT & TOOLS

Since SDL students are typically working on large scale projects that span many quarters, they are exposed to various tools for time tracking, version control, issue tracking and defect logging.

Reliable and complete measurement data is critical to effective planning and process improvement. If the process of gathering this data and analyzing it becomes burdensome, then it will not be done [8]. In SDL, two internally developed tools: Leia (Laboratory Information Engineering Archive) and FAST (Friendly Assessment Support Tools) are used for data management that support project planning and tracking based on defined measurement frameworks. The Leia tool is based on TSP and currently has functionality to manage components and tasks, edit cycle plans, log time and defects, display time logs, track earned value and reporting of quality metrics. FAST is used to track total lab time (task hours + overhead hours) while only task hours are entered in Leia.

An open source version control system called CVS is used for configuration management. Mantis, another open source product, is used for defect/issue tracking. Additionally, there are several tools that are available to support SE practice. Enterprise Architect (EA) is used in the lab as a modeling tool.

Different projects require the use of different technologies and therefore use different integrated development environments (Eclipse, VC++, VS.NET, etc.). Several testing frameworks are also available for students. Few projects also use proprietary software packages that are typically provided by the client. Additional software and hardware requirements for a particular project are handled on a case by case basis by the instructors in consultation with the project’s client.

We have two dedicated web servers for the SDL – one is the SDL’s production server (opal.msoe.edu) and the other is its pre-production server (garnet.msoe.edu). Version control for code and documentation, defect tracking system (Mantis), Leia, FAST, and client applications are hosted on the production server. Each team maintains a web site for their development server which is also hosted on the production server. The team web site contains all project documents (unless protected by a specific non-disclosure agreement), status updates, cycle reports, meeting agendas and minutes, bulletin items, risks and release logs. Each team must keep these web pages current to apprise the clients and the instructors of the current state of their project. SDL members are provided shell access to the production server to make their updates.

The pre-production server (garnet) is configured exactly like the production server (opal) and is used as a beta site for applications before they are deployed. Students do not have access to production databases on opal, but do have the authority to manipulate test databases on garnet. Deploying an application to garnet before finally deploying it on garnet, helps the students test their installation scripts. Deployment of an application to the production server is typically done by an instructor (as the students do not privileged access to the machine) using the installation scripts developed by the students.

Students also have access to a refrigerator, shredder, and a conference telephone from the lab.

CHALLENGES

Even though the SDL for the most part has been a positive experience for both the students and instructors, it has had its fair share of challenges. Some of the challenges that we are currently facing in the SDL, especially in executing the original pedagogical philosophy of the lab are:

- **Time Spent by the Students in the Lab** - Analysis of the time logs submitted by students for the last two years shows that an average student spends about 50% of the budgeted time (i.e., 5 hours) in the SDL; for most students the time is spent on the development projects with no significant staff deliverables. Of the actual time spent in the SDL, only half is spent on task-related hours, the rest being overhead. In other words, a typical student spends about 2.5 task related hours in the SDL per week, which is not enough to make any kind of “real” progress on the product. Deadlines in other classes, unstructured environment of the SDL, no identical assignments/deliverables across the lab, and time wasted in context switching are some of the reasons that are offered by the students for their performance in the SDL. Many students have suggested we schedule longer lab sessions and ensure that the instructor is available. With such a scheme, the team would not have to work around individual scheduling issues. They also believe that less time will be spent on context switching. While the suggestion has merit, academic budget constraints limit instructor time for a 3-credit course and overall class scheduling requirements restrict flexibility.

- **Few Success Stories** - Students are expected to spend about 10 hours/week in the SDL. Even when this goal is met, a student spends in a month as much time as a working professional would do in a week. For this reason, the projects in the SDL move at a slow pace and it is very hard to sustain student motivation and client interest. In the last three years in the SDL, two projects have been brought to a close. One was delivered complete and the other was cancelled. After five years in the lab, the cancelled project was still behind schedule and the client judged that their window of opportunity had long since passed. Leia (the lab’s in-house planning tool) was finally deployed with limited functionality in the lab in winter 2005-2006. This was achieved after some components were developed by instructors and increased project monitoring was provided. The current Leia team continues to deliver additional functionality.

- **Lack of Mentoring** - The classes in the SDL sequence are scheduled to meet for four hours each week (typically for two 2-hour sessions). The remaining time that a student spends in the lab is unscheduled. When there were five development teams in the SDL and only one instructor to go around, a maximum of 48 minutes per group was
available which is insufficient for any kind of active monitoring and mentoring. In practice, the instructor usually spends about 30 minutes with each team and uses the rest of the time to handle lab administration issues and conduct lab wide meetings. With two instructors in the lab, the situation has been somewhat alleviated.

- **Choice of projects** – We have always tried to get SDL projects from external stakeholders. Because of the slow development pace, these projects also tend not to be “on the critical path” for the sponsor. Since the stakeholders also tend to be very busy, they may not interact with the students and display their enthusiasm for the project on a continual basis. The students tend to interpret this lack of participation as meaning that nobody is waiting to use the product that they are working on. This tends to be a huge de-motivator.

- **Unfamiliar Technologies** – In some instances, the projects that are being developed in the lab are using products and technologies that the instructors as well as the students are not fluent with. In that case, it becomes harder for the students to get help when they need it.

- **Grading** – As in other courses, the grading criteria for the SDL sequence are made available to the students on the first day of class. Unlike other classes, students have a much harder time calculating their grades based on the feedback that they receive from the SDL instructors and hence are frustrated. Whether we like it or not, grades continue to remain one of the biggest motivators for all students in the classroom.

## REWARDS

Time and again, students cite SDL as one of the most valuable courses they take in the undergraduate career at MSOE. Even though, it seems like a frustrating experience for the students when they are enrolled in the SDL, they seem to see the value it provides when they work on their internships/summer jobs between their junior and senior years. Most students come back after the summer and report with surprise that the “real-world” is indeed similar to the SDL’s structure. They realize that they experience success in these jobs early on because of their familiarity in how to handle and operate in such an environment. Most employers of our students seem pleased with what SDL has to offer and become involved (and stay involved) with MSOE by becoming stakeholders for an SDL project.

## OPPORTUNITIES

The presence of SDL in the curriculum, and the underlying infrastructure and facilities provided to support it has also presented us with many opportunities.

While enrolled in the SDL, students consider it their second home. They use these facilities to do homework, practice their presentations not necessarily for SDL and to hang out with their friends. This time spent together in the SDL builds a sense of ownership, community and pride within the SE students. The conference room is also regularly used for program meetings, industrial advisory committee meetings, senior debriefing and freshman orientation. We are working on the logistics for making the lab more accessible to SE students who are not currently enrolled in the SDL.

The SDL is also used as a recruitment tool. We give tours of the SDL facilities to prospective students and their parents and talk about the unique opportunity that this lab experience presents. It appeals to the prospective students because of the technology it houses and to the parents because it resembles their workplace.

The SDL has also been used to host workshops and corporate training sessions being led by MSOE SE instructors. The SE students have also used the lab for a “SE party” where all the SE students got together for fun, food and an evening of video games.

## CONCLUSIONS

In conclusion, we would like to say that the SDL at MSOE is a unique experience which provides the students a unique opportunity to work on real life projects in an academic setting. This experience, though challenging at times, is well received by the students, their potential employers and the instructors.

## REFERENCES


