

Reflection of my Academic Experience

By *Matthew Ring*

It's a good feeling to look back at my four and half years of college and see just how much information I crammed into my brain in that amount of time. At first, it seems like a blur of events and topics all thrown together, but after a little while, certain concepts appear out of the mist.

Learning Effective Software Design:

One of the most useful courses I took (and later served as a Teaching Assistant) was over object-oriented analysis and design. This class had a two-fold objective: To introduce users to OO programming using Java; and to start developing a project using iterative development and the Unified Modeling Language (UML). This class gave me a lot of insight about taking an existing problem and making sure that any proposed solution would be the most beneficial and cost-effective (in terms of labor and price).

Before this course, most of my programming and engineering classes spent little time on solving a problem efficiently. As long as the program worked, nobody cared how bloated the code was. At the time, my projects were small enough that any inefficient solutions didn't hinder the performance of the system. Now, I was forced into really thinking about the best way to solve a given problem, and I've seen the benefits of this approach in my programming projects since.

Relating Concepts to Real-World Experiments:

It's one thing to be taught mathematical integration. It's another to construct an op amp circuit and see a waveform showing integration of a signal on the screen. It tends to give more respect toward the concept when it can be reproduced in a real-world environment.

Throughout my college career, I've had to perform several labs, from high- to low-level programming to circuit design and analysis. Some labs were interesting; others were a bit mundane and tedious. But all of them help integrate concepts said in a lecture to real-world uses. In addition to the example above, one of my labs required me to write a simple shell program in C that ran on Linux and emulated a command prompt, such as file copying and process-calling.

Utilizing Industry Development Tools:

My first internship introduced me to a lot of development tools and applications that I had never previously been exposed to. In college, courses take an entire semester *telling* you about a development tool or a range of concepts; in the workplace, you spend eight

hours a day, five times a week, *working* with a development tool and you pick it up pretty quick (or become dead weight fast).

In future job interviewing, I have become less concerned about knowing a particular development tool than understanding the basic fundamentals of the job. I can learn quickly enough how to use Visual Studio.NET to build an ASP.NET application; but it's more important that I know fundamental programming techniques and other concepts, like effective user interface design and current web standards.

Working Together to Make a Better Product:

In my upper-level courses, it was common that a project be done in a group of three or four students. It wasn't an unreasonable assumption since real-world companies can take weeks or months to put out a commercial product at 40 hours per person per week.

Some groups were assigned at random; others were assigned by whoever lucky soul was sitting close to you or if you were friends with anybody else taking the course. For me, I had the fortune (and misfortune) of handling both situations.

Senior Design was my first team experience developing a complete project from start to finish. I hear many students condemning senior design, calling it a waste of time. For me, though, it was an excellent experience and a great learning tool.

My group had a tough situation where we lost a team member early on, bringing our size down to three for the two-semester project. However, we were very motivated and worked so well together that we were able to delegate out responsibilities to make up for the lost person. We disagreed at times, but were always able to find a compromise to benefit everyone.

In addition, we made an emphasis on effective communication to maximize the use of our time and keep track of what each person was doing. Without this, it is possible that we would have each gone off on our own tangents and would have dissolved the tight formation of our project.

Because our project dealt with some technologies that we were not familiar with (e.g. JDBC, efficient database schemas), we each fed off each others' experience, both academically and from internships, to come up with the best solution. It also gave us experience working with a non-technical client and finding a way to communicate effectively with him to find what features were most important to the user.

Keeping Textbooks for the Long Haul:

It's a bit ignorant to take courses in an ever-changing degree for five years and expect to know everything needed by graduation. More courses would pop up from semester to semester, offering information on new topics such as Linux and the Microsoft .Net Framework.

I believe it is important to hang on to some of the textbooks that I think would be good for referencing later on in my career. I don't expect to know everything about a programming language. I do expect that experience will give me the knowledge that, given a task, I will know if I *can* do something, and my reference material will be there to help me on *how* to do it.

Knowing When to Act Like a Grown-Up and be Responsible:

I remember my internship supervisor telling me how he appreciated that I acted like a professional when I needed to be one. That gave me a great sense of pride that I was able to joke around with co-workers when it was appropriate, but could turn around and get down to business when things needed to get done.

I think it's important for employees to have a balance of fun and professionalism in the workplace. College advisors always tell you that you need to be professional in the workplace; but at my internship, I soon realized that it's impossible to be serious all the time because then the job becomes mundane and boring.

There has to be a balance of professionalism and enjoyment in the workplace; otherwise, the turn-around of employees becomes too great.