Why Educators Need to Team with Industry Professionals in Software Development Education

Dr. Gregory Kulczycki, Virginia Tech

Dr. Kulczycki has extensive experience in research and development both in academia and industry. He received his doctorate from Clemson University in 2004 and began working as a professor at Virginia Tech shortly thereafter. In 2011 he went to work for Battelle Memorial Institute as a cyber research scientist, while continuing to be involved in teaching. He is currently back in the computer science department at Virginia Tech as a professor of practice, where he teaches, designs courses, and develops online content for Virginia Tech’s highly-regarded Master of Information Technology program. Dr. Kulczycki has various publications on topics including formal specification and verification, web services, and software reuse. His interests include object-oriented programming, software specification and reasoning, design patterns, and online learning.

Dr. Steven Atkinson, Virginia Tech

Steven Atkinson started his research career at the Software Verification Center at the University of Queensland, in Brisbane Australia. He worked as a postdoctoral research assistant professor at West Virginia University, before transitioning to industry for the last 18 years. He has worked at enterprise document management, healthcare and high performance networking startups. He was one of the first employees at LinkedIn, and has worked for 9 years for Netflix, where he currently works for the DVD division. He also works as a distance instructor at Virginia Tech in the Computer Science department.
Why Educators Should Team with Industry Professionals in Software Development Courses

Abstract

One author of this paper (Dr. Kulczycki) is a professor at Virginia Tech and the other (Dr. Atkinson) is a Silicon Valley web developer. This paper presents our experiences in co-teaching a web application development course for a graduate-level information technology program.

It seems obvious to say that students benefit by having access to industry expertise when taking a course on software development. An industry perspective on any academic subject can be helpful, and as software development is such a practical side of software engineering, students often take these courses with an eye toward developing software professionally. Professional developers can not only answer student questions about how something is done in industry, but they can make the student understand why a topic is relevant.

Educators tend to have a good grasp of the capabilities and limitations of their students. They are good at choosing course materials that meet their course objectives, and they know how to assess students on whether those objectives have been met. Professionals know how things get done in industry. They understand the trade-offs and compromises that must be made to get a software product out the door. Their knowledge comes from other developers and from years of practical experience building real-world systems. When the educator and the industry professional are teaching as a team – both invested in the outcome of the course – the results can be transforming.

This was our experience in teaching a course on Web Application Development for an online Master of Information Technology program. The students in this program are typically working professionals in an IT-related field, and their programming experience varies widely depending on whether they come from the business side of IT or the technical side. All students taking the course are required to have a basic introduction to Java. The course is completely online, and student-teacher interaction comes primarily from Q&A discussion boards (Piazza) and one live Q&A session per week (WebEx). The course revolves around a semester-long project in which students develop a mini e-commerce web application complete with the design and implementation of the web interface, the database, and the application business logic.

In this paper, we talk about how the course evolved when the developer joined the educator to teach the course. We focus on six important facets of the experience: (1) the initial conditions that allowed the collaboration to be successful, (2) the benefits that each party brought to the course, (3) the impact that online learning had on the collaboration, (4) how the development of custom tools improved the course, (5) the introduction of industry best-practice to the core project, and finally, (6) the challenges that face us as we continue to improve the course and apply our experience to other courses.
Introduction and context

Most courses in a practical discipline such as software engineering will benefit from the involvement of industry professionals, even if that means simply inviting people from industry to talk to students once or twice during the semester about what it is like to apply software engineering principles to developing commercial software. Courses that focus on software development can particularly benefit, since the students taking these courses often have some interest in professional software development.

The level of involvement that industry professionals take in a course can vary widely. Most computer science and software engineering programs offer a seminar course in which they invite both outside researchers and industry professionals to speak to students. Some departments have programs that assist students in collaborating with local industry during capstone projects. Current or retired industry professionals may teach a course as an adjunct professor. And some departments have full-time instructor positions in which applicants are encouraged or required to have a certain number of years of practical experience [3].

In this paper, we discuss our experience with academic-industry collaboration in the classroom for a course on Web Application Development. While we are convinced that our collaboration was – and continues to be – a successful one, we understand that it occurred under specific conditions that may not hold for many university courses. Most of these conditions benefited the collaboration.

1. The course is a *graduate-level* course and is geared towards *working professionals*
2. The course is taught *entirely online*, with recorded lectures, discussion boards, and a live Q&A session once a week
3. The course includes a *semester-long project* that accounts for about half of the grade

We should also mention that even though the course was a Computer Science course, it was only open to students who were enrolled in our Master of Information Technology (MIT) program. The MIT program is a collaboration between the Business School and the College of Engineering – specifically, the Computer Science Department and the Electrical and Computer Engineering Department.

As the students in the course were graduate students, they had a certain level of maturity that undergraduates might not have. And as many were also working professionals, they brought a certain level of professionalism to the course as well as an expectation that they were getting their money’s worth. The fact that they were MIT students rather than CS students meant that they were coming to the course with a wide variety of backgrounds. Some may come from a primarily business background and have very little programming experience, and others may even have prior experience in developing web applications. The fact that some students in the course had web development experience was a significant motivator for the educator to consult a
professional in the first place. All students must have taken the prerequisite for the course, which is an object-oriented programming course that uses Java.

The online nature of this course was so significant and so potentially different from traditional courses that we devote an entire section to it and its effects on the collaboration. It certainly made it easier to recruit a working professional to co-teach the course as it provided maximum flexibility for their schedule. Lectures (done by the educator) were pre-recorded. Interaction between the students and the instructors was achieved by an online discussion board and one weekly Q&A session. The response time for questions on the online discussion was typically 24 hours or less, and the Q&A sessions were live online, but attendance was not mandatory. The sessions were recorded.

For their semester-long project, the students had to design and implement a mini e-commerce web site – a bookstore site that could display different categories of books from a database and allowed users to add and remove books to and from a shopping cart. On a checkout page, users would fill out a customer information form that was validated from both the client-side and the server-side. “Purchasing” the books would update a database. As basic knowledge of Java was a prerequisite for the course, JSP and Servlets were used for server-side implementation. An SQL database (MySQL) was used to store information. Frameworks were generally avoided. The students submitted (for a grade) portions of the project approximately every two weeks. Each portion built on the last until at the end of the semester they submitted their final web site.

Before the professional web developer joined the course, the educator relied heavily on the NetBeans E-Commerce Tutorial [8] to demonstrate how the project for the course should be implemented. In fact, most sections of that tutorial were required reading. A significant motivator that led the educator to seek out professional assistance was the concern that the design of the Affable Bean Project (the web site that was developed by following the NetBeans E-Commerce Tutorial) may not reflect industry best practice. The tutorial is very well written and covers the major objectives of the course, but it is somewhat dated (it was written in 2009), and a portion of it emphasizes EJB (Enterprise JavaBean) technology, which is not as prevalent in industry as some other technologies. When the educator and professional were discussing ways to modify the Affable Bean Project to bring it more in line with industry best practice, the professional suggested employing a version of the DAO (Direct Access Object) Pattern using Java instead of relying on any frameworks. This was in line with the educator’s wish to rely less on frameworks. Details of how the Affable Bean Project was modified by the professional are given in a section below.

Finally, before we delve into these topics, we would like to acknowledge that there can be challenges in teaming with an industry professional above and beyond those faced when two people who think highly of themselves try to do something together. Specifically, the challenges we are thinking of relate to tools and technologies. Simply put, industry professionals tend to want to use or teach about tools and/or frameworks used in their current job. Obviously, this can
be problematic. Industrial-strength tools can have a steep learning curve and can distract students from their learning of basic development principles. Furthermore, specialized frameworks also have a steep learning curve and are often overkill when teaching software development. Even the author of this paper who is the industry professional believes that a good developer puts good software engineering principle and patterns first and should not rely as much on specific tools or frameworks, but it is human nature to promote what we are comfortable with. As an educator, if you meet a professional who seems to favor tools or frameworks over principles and patterns, it is probably best not to pursue a collaboration with that developer [4].

The remainder of this paper is divided into two broad sections. The first covers several factors that we feel were important to the success of the collaboration, including conditions agreed upon before the course began, the benefits each party brought to the course, and the impact that online learning had on the collaboration. In the second part, we give two examples of how industry expertise improved technical aspects of the course – by facilitating custom tools development and by introducing industry best-practice into the core project. Finally, we look at challenges that face us as we continue to improve the course and apply our experience to other courses.

Prerequisites for a successful collaboration

This section is given in two parts. First, an account is given by the educator of how the collaboration came about. Second, we discuss several factors that existed before the course began that we believe helped lead to a successful collaboration.

What follows is the account of the collaboration from the educator’s point of view.

In the summer of 2016, I was tasked with redesigning a graduate course called Internet Software and bringing it up to date. The course covered a variety of topics, and I wanted to focus on something that would be relevant to the students in our program, who were mostly working professionals. I chose to focus on Web Application Development, and my idea was to have a semester-long project in which students would create a mini e-commerce web site. When I was challenged by one of the instructional designers from our university to come up with a goal for the course, I thought of something practical: A student completing the course should know enough about web development to impress a potential employer at a job interview.

That same summer I was on vacation in California and decided to stop in to see an old friend and former colleague who had been working in Silicon Valley for over a decade. Because I had been working on the course and because he was a web developer by profession, our conversation focused a lot on the course. One interesting thing I remember him saying was that he was at a point in his career where he was looking to give back to the development community. At that point, I don’t think either of us were thinking in terms of collaboration, but I did tell him that I would keep him informed about how the course was progressing, and
he told me that he would be happy to volunteer to give an online Q&A session about professional web development to my future students.

During the first semester I taught the class, I realized that the project had a weakness – namely, me. We were using the NetBeans E-Commerce Tutorial as a guide to our project, and I had no problem explaining the concepts behind the code or how the code worked. But when I was asked about why we were doing something one way rather than another I struggled to give a good answer. My fallback for questions when it came to frameworks was that we avoided them so that students could learn the principles behind them, but even that was not entirely satisfactory as the project relied on EJB technology to interface with the database. What I really wanted was someone with authority to tell me how our project would be designed using industry best practice.

Though I thought my friend in Silicon Valley might be able to give me guidance on this, I still hesitated to ask him about participating in the next semester’s course as a distance learning instructor. First, because I was not sure if our views were entirely compatible, and second, I doubted that a successful Silicon Valley developer would want to devote time to teaching. When discussing the project with him, however, one of the first things he suggested was modifying it so that it used a DAO Pattern – implemented entirely in Java – rather than EJB technologies, thereby reducing reliance on frameworks. As for the time commitment, I was content to let him make that decision for himself, but only after I was convinced I could handle things alone if necessary. As it turned out, we both ended up spending more time on the course than we had hoped, but the result was satisfying.

We feel there are several factors that contributed to a successful collaboration in our case.

1. Academic goals and objectives take priority over everything else
2. Core learning principles are emphasized over specific tools and frameworks
3. Educator and industry professional share the same basic vision
4. The industry professional is invested in the course

The first point here is the most important. In fact, the others really follow from this: that academic goals and objectives must take priority. Despite the fact that many of our students are working professionals and have a unique appreciation of expert industry experience, what they signed up for and what they are paying for is a university-level course on web development. And if that course fails to deliver what it promises, the reputation of the program and the reputation of the university is what is going to suffer.

University courses tend to do an excellent job of teaching fundamental principles to students, and that focus should not change. When someone joins a new company to participate in software development, one of the first things they must do is get up to speed on the tools and technologies that are used by that company. There is a non-trivial learning curve for doing that. It is essential that the student’s time – particularly in a one-semester course – is mainly devoted to learning
development principles rather than specific tools or technologies. This is particularly true when it comes to web development, as there are so many different frameworks being used in industry. In our case, when it became clear that both the educator and the industry professional felt the same about minimizing the reliance on frameworks, the collaboration was imminent.

Where we did use specific tools, we tried to minimize their impact. For example, the industry professional was comfortable with IntelliJ IDEA as an IDE. Most IDE’s have similar functionality, and students were already familiar with other IDE’s, so the learning curve was not significant. Also, we decided to use Gradle to set up the project. We did this by simply providing a short `build.gradle` file to the student and going over what each line meant. After that, there was barely any mention of it.

The fact that our vision for the course was mostly compatible meant that we did not spend a lot of time debating what was best for the students. We even had a more-or-less similar view on what role the industry professional would take in the course. In our case, the professional focused on revising and updating the project, as well as fielding student questions about technical aspects of the project in the online Q&A discussions. The educator created video lectures, led the online Q&A sessions (often by walking through the beginnings of the projects), and handled the grading and administrative aspects of the course. We will talk more about these tasks in the next section.

Finally, we think it was helpful that the industry professional was so heavily invested in the course. Both financially, for he was getting paid; and emotionally, for he was interacting with students, sometimes on a daily basis. If the task of the professional would merely have been to rewrite the project according to industry best practice, or to lead a Q&A session once a semester, the results of the collaboration would not have been as impactful.

**Educator vs industry professional**

Both the educator and the industry professional have expertise they can contribute to a software development course. Key qualities that an educator brings to the course include the following.

1. The educator is good at meeting learning objectives
2. The educator has a good sense of student capabilities
3. The educator knows how to assess student learning

Since educators are the ones who create course objectives, it is their responsibility to make sure that those objectives are met. As stated in the previous section, the goals and objectives of the course must take precedent over everything else. That means that the industry professional must ultimately defer to the educator when deciding what to include or what not to include in the course. In practice, course objectives are broad enough to allow significant flexibility as to how
specific objectives are met. For example, in the web development course, one of the broad objectives reads:

*Apply server-side web technologies such as Java Servlets and JSP, in the creation of dynamic web content.*

Even though the designers knew they wanted to teach the course using Java in the foreseeable future, they included the phrase “such as” to give future instructors maximum flexibility.

The educator will also be better aware of what the students can and cannot handle. This is particularly relevant when it comes to deciding on course material and creating assignments. As our course was mainly project driven, much of the course content needed to be updated as the project was modified. Though we agreed that the industry professional should redesign the project from the *NetBeans E-Commerce Tutorial* to reflect industry best practice, not all of those changes were incorporated in the project that the students had to do. There was always a discussion on how to balance the most relevant aspects of the project with what students could accomplish in the time frame they were given.

The educator is also going to better at assessing student progress, that is, in knowing how to grade the students. This fact led to an interesting result in our situation. For courses in our program, we are permitted to hire a distance learning instructor (DLI) when the enrollment exceeds a certain number of students. DLIs may be adjuncts or graduate students (GTAs), and they often help with grading. In our case, the educator decided he would grade the projects himself and use the professional – who came on as an adjunct – for modifying course content and fielding student questions.

Important qualities the industry professional brings to a course include the following.

1. The professional knows how development is done in the real world
2. The professional can tie learning principles to real-world practice
3. The professional is familiar with solutions based on industry culture

One obvious way that a professional developer can help a project-based course is to ensure that the projects in the course reflect an industry-level standard. This is certainly something the professional did for our course. But an added benefit of having a professional developer on board is that they understand development environments and can therefore aid in creating an environment – along with custom tools if necessary – that make it easy for students to set up and submit projects, and for educators to grade them. We will discuss this further in a later section of this paper.

Professionals can relate the material that students are learning to industry practice. This is one of the most important reasons for collaborating with someone in industry. The answers that an industry professional gives to a student question carries so much more weight than the answer an
The educator can talk about design principles, published patterns, and refer students to different resources; but the professional can give them real-world examples, practical advice, and let them know what works and does not work when it comes to commercial software.

Professional developers are more likely to use a methodology explained to them by someone they trust rather than use something they read about in a journal. Both sources are important, but educators – especially those who are researchers as well – tend to put more emphasis on published sources. Educators and professionals will need to find an appropriate balance between both approaches. To give just one example, when the professional in our course revised the project so that it used the Direct Access Object (DAO) pattern instead of EJB technologies, the implementation he used differed somewhat from published explanations of the pattern. The model classes in the published version were mutable, while the professional made them immutable. We discussed this, and both of us agreed that immutable classes were preferable. Therefore, students were assigned to read an account of the DAO pattern in one of their texts, but the educator also modified his video lecture to illustrate the difference between the two approaches.

There is one benefit to the professional that we did not list, but it still bears mentioning: The professional developer has friends who are professional developers. In our case, the professional developer mainly dealt with server-side technologies in his work. But in part of our course we wanted to talk about website design, so the developer enlisted the aid of a GUI specialist who owed him a favor and we were able to give the students a template that they could use to help them design their own GUI.

**Impact of online learning**

In this section, we talk about how the online learning environment in our course facilitated the collaboration. During a typical week in our course, the student will do the following.

- Listen to the recorded lectures
- Do the reading assignments
- Take the reading quiz
- Work on the current project assignment
- If you have questions about the project, ask them on Piazza
- Attend (or view later) the live online Q&A session

The course is not only online but it is also asynchronous. Each week a new learning module is introduced, and the students typically have that week to work on the assignments. The educator creates the recorded lectures, which are mostly slide presentations with a voiceover. The readings are usually available online. The reading quiz is taken via our Learning Management System (LMS) and is graded automatically.
The weekly project is a portion of the mini e-commerce website they are developing. It includes both functionality and design requirements. Students develop the project using IntelliJ IDEA on their local machines. Each student has a dedicated database set up for them on our remote server. If they have any questions about the project, they ask them on the online Q&A forum (Piazza). Both the educator and the industry professional field those questions. In practice, the professional fields questions about project implementation, while the educator fields questions about project requirements. When students are happy with their project, they upload their WAR file to the server and the educator grades it (project grading is not automated).

Live, online Q&A sessions are held one evening per week and only last about an hour. Attendance is not mandatory, but we do require students to view the recording if they cannot attend. Even though they are called Q&A sessions, we almost always have something planned. A common occurrence is that the educator will share his screen and will begin going through the project on his computer. The industry professional also attends the Q&A session and will chime in whenever he wants to make a point.

We have found that certain aspects of this online learning environment are beneficial to a collaboration between the educator and the industry professional.

1. It allows the industry professional maximum flexibility in terms of time and location
2. The online Q&A forum provides daily interaction between students and the industry professional
3. The live Q&A sessions provide an opportunity to reach a wide variety of students

The first point is critical to our course. The educator is located on the East Coast and the industry profession is located on the West Coast. Without some sort of online technology, the collaboration could never take place. In addition to location, the fact the industry professional only needs to keep one hour a week free for the class (to attend the live Q&A session) makes time management much easier than for a traditional course.

The online Q&A forum is one place where the presence of an industry professional is invaluable. At the beginning of the semester, we make a commitment to respond to questions posted to the Q&A forum within 24 hours. The answers that the industry professional can give to a student when it comes to the design or implementation of the project far surpasses any answer the educator could give, particularly in terms of detail and context. The professional can not only give a good how answer to a question, but they can give a more convincing why as well. It may take some time for the professional to learn how to respond to a question while still encouraging the student to find the complete solution for themselves, but this is a skill that is quickly learned. We cannot overstate the benefit of students having daily access to a professional developer in this way.

The live Q&A sessions are something we have been slowly improving upon by using trial and error. Currently, what seems to work best for us is the following. The weekly module – and
therefore the project – is assigned on a Monday. The live Q&A session is held on Wednesday at 8pm EST. During the session, the educator walks through the first part of the project, making comments as he goes. Note that the educator is doing this using his web site design (everyone’s will be slightly different), so students still need to apply the actions to their own project. The industry professional follows along, and when he feels he can elaborate on something the educator is saying, he does so. The effect this seems to have is that even students who are struggling get the benefit of seeing how to get started, and even students who are more advanced get the benefit of hearing an industry professional give them expert insights. One thing the educator has learned while doing these partial walkthroughs is that they do require preparation to be effective. Complete them ahead of time and know when to stop and ask questions of the students; and also know when to stop and ask questions of the industry professional.

Before we leave this section, we should point out that even though our course takes place entirely online, it would not be difficult to get these same benefits in a face-to-face course, provided the course incorporates an online Q&A forum and the educator can bring the industry professional into the classroom once a week via online videoconferencing.

**Custom tool development**

In the first version of the course, before the professional developer joined the team, all versions of the projects were graded by having the students submit WAR files to the instructor. The instructor graded them by running the files from within his version of NetBeans on his personal computer. The process was cumbersome and error prone. The intention was always to have a dedicated server which would not only store the SQL databases for each student but would also allow each iteration of each student project to reside on the server for the instructor – and other students – to see.

This became much easier to do once the professional joined the course, not because the professional did the work. On the contrary, the educator not only administered the server, but he also developed a web application that permitted students to upload their WAR files directly to the server. However, the fact that the educator had a professional developer to consult with, who was easily accessible for answering questions, troubleshooting, and debugging code problems gave the educator a level of confidence and comfort in tackling these issues that was not available before the professional joined the course.

Even before the professional joined the course, the SQL databases for the projects resided on the course server. Each student had their own database. The decision to eliminate the need for an EJB container from the project greatly simplified the task of putting student projects on the server. We no longer needed GlassFish (or another EJB container like Firefly), we could simply install Tomcat, which had more online support and was more familiar to the professional. Installing Tomcat and placing student projects in the webapps directory turned out to be fairly
straightforward. But when some students began having problems related to SQL connections (their SQL databases were on the same server), it helped that the professional was there to troubleshoot and show students how to clean up connections when they were no longer being used.

A more sophisticated project involved building a web application on the course server that allowed students to upload their WAR files directly to the server. The application needed to be secure, it needed to validate student names and the names of the WAR files they intended to upload, and it needed to upload the WAR files and place them in Tomcat’s webapp directory. Having this application greatly streamlined the submission process and allowed students fast access to their projects once they were uploaded (we asked that they wait a few minutes before trying to view them). Furthermore, putting the projects online made it much easier for us to implement peer reviews for selected projects, since all students had access to all projects.

The next tool we have discussed is one that could semi-automate the grading process for the functional requirements of the projects. This one has not been implemented yet, but the framework is there. It is likely that these tools would eventually have been developed even if the professional had not joined the team, but they would have taken significantly more time and taken away from the educator’s ability to focus on the teaching.

**Applying best practices to the project**

As we mentioned previously, the industry professional was primarily sought out to apply his knowledge of how commercial software is developed to the semester long e-commerce project assigned to the students. The class project closely followed the well-written and publically available NetBeans E-Commerce Tutorial, so rather than work on the class project directly, we decided that the professional would modify the Affable Bean project from the NetBeans Tutorial, and the educator would then bring the class project in line with those modifications. In this way, the professional had maximum flexibility with the changes he could choose to make to the code from the Tutorial, and the educator still had control over the project, ensuring the it stayed in line with student abilities. A side benefit of this approach was that the professional could post the revised code publically, since the Affable Bean code is published under an open-source BSD license. The revised project is called Simple Affable Bean and is described at the web site simpleaffablebean.info. The following text is taken directly from that web page and details why the professional felt that Affable Bean needed to be rewritten. For context, recall that the Affable Bean project uses EJB (Enterprise Java Bean) technologies.

**Why Rewrite Affable Bean?**

In my experience, hardly anyone uses the EJB stack as it was meant to be used, with session and entity beans, data sources, services and the rest of it. I’ve found that some of the EJB
technology have too much abstraction. This means the frameworks are seductively easy to use. However, they tend to lead to:

- code that is difficult to reason about and
- code that is less amenable to change (i.e. the developer has diminished ability to make changes easily when business context changes).

So, when starting the rewrite of the original project, the key goals were:

1. to require a servlet container rather than an EJB container,
2. to maintain all functionality in the original project.
3. to avoid the use of high-level frameworks to better demonstrate the basic building blocks of web application development.

While rewriting the code, it became apparent that there were many improvements to make. (In fact, there are always more improvements to make, even now!). We decided to focus on a few major concerns: having a better code architecture (separating business logic, database logic from controller logic), using a low-level no-framework database layer, and using the View Model design pattern to simplify view templates.

An important thing to notice about this statement is that, in some sense, you can replace EJB with almost any sophisticated web framework and come up with a similar rationale for rewriting the project. Frameworks provide a level of abstraction and can help you create web applications faster provided you have a good understanding of the framework. However, if you need to debug a problem in code that relies on a framework, you often need to have a great – not just good – understanding of the framework and how it integrates with the core language. We mention this because it recalls one of our first observations – that the educator and the professional should have the same basic vision. In this case, we shared the vision of minimizing dependency on frameworks and showing students how to develop as much as possible using the core language.

We see at least four broad categories of how having a professional developer on board can positively impact a major course project.

1. The professional has expertise in industrial-strength tools
2. The professional has access to a network of other professionals
3. The professional has a deep understanding of common patterns and techniques
4. The professional has knowledge of the latest patterns and techniques

In the following subsections, we give examples for each category of modifications made to the course project that illustrate both the introduction of industry best practice by the professional and the input from the educator regarding how to introduce those changes to the students.
Expertise in commercial tools

This benefit may seem a bit at odds with the warning we gave earlier that professionals can become overly attached to tools and frameworks that they use frequently. We do not, however, want to rule out a tool or framework simply because the professional has a bias for it. As an example, we made the switch from the NetBeans IDE and the GlassFish application server to IntelliJ and Tomcat. The move from GlassFish to Tomcat was an obvious one. As we had already decided to move away from EJB technology, we did not need the complexity of a full application server like GlassFish. Tomcat is used much more often in industry due to its simplicity and robustness, and it seemed to have much more online support. The choice of IDE was a bit more arbitrary. The three main IDE’s for Java – Eclipse, NetBeans, and IntelliJ IDEA – are very similar, and knowing one of them, it is not difficult to migrate to another. The professional was comfortable with IntelliJ, while most of our students had experience with Eclipse. As we were already asking the students to transition to a different IDE (NetBeans) in the original version of the course, we did not think it would be a problem to ask them to transition to IntelliJ instead. The fact that the professional would be able to answer student questions on IntelliJ more easily than another IDE supported the decision. Finally, we were in the process of creating a Mobile Application Development course that employed Android Studio, which is built on IntelliJ, so the transition could be further justified.

Access to a network of other professionals

A major benefit of collaborating with a professional developer is that they know other professional developers. In our case, the professional was able to recruit one of his colleagues to develop a template for the web page layout of the project using HTML and CSS (but not JavaScript – we did not want to introduce that until later). For a commercial product, the web designer would have based the layout on HTML tables and inline blocks for compatibility with the largest number of browsers. However, having seen the difficulty that students had with tables, floats, and inline blocks in the original version of the course, the educator asked the web designer to use flexboxes for the layout. They were much simpler to use than tables and floats, but they were a relatively new addition to HTML. Web design was important, but it was not a major portion of the course (one module out of twelve), so in our case, the decision to use flexbox considerably simplified page layout for the students.

Another issue that arose with the web designers page layout was that the web designer used classes everywhere in his CSS files instead of identifiers. This is common in professional web development because classes are reusable within a page. But professional web designers have a sixth sense for consolidating elements in classes and it was not always obvious to the educator how to explain the rationale for the consolidation. Therefore, during the partial walkthrough of the web design portion of the project, the educator preferred identifiers over classes. We should emphasize that the page layout templates created by the web designer were not something that the students were meant to use and then modify; the students were expected to create their own
layouts, borrowing elements from the professionally-designed page layout templates as necessary. We stressed to the students the importance of fully understanding any code they included in their projects.

**Deep understanding of common patterns**

A professional developer who has been working in industry for years understands common design patterns well and knows how they can be fine-tuned to without compromising their essential benefits. Replacing EJB technologies with a DAO (Direct Access Object) pattern was the first and perhaps most important change that the professional web developer decided to make to the project. The change removed the dependence of the project on EJB technologies and actually eliminated a class of hard-to-debug student errors in the original version of the course that arose when misusing the flush method with an entity manager during transactions [5].

![EJB-based design](image_url)

**Figure 1** EJB-based design
Figure 2 Custom DAO pattern

Figure 1 shows the EJB design and Figure 2 shows the custom DAO design. Both designs have two main types of classes: (1) classes that represent data from tables in the database and (2) classes that represent CRUD operations (create, remove, update, and destroy) on those tables. In the EJB design, these classes are called entity class and session beans respectively; in our DAO design, these classes are called model classes and DAO classes. One advantage of the EJB classes is that they can be generated automatically be NetBeans, though typically developers will modify them a bit.

Another benefit of EJB classes is that many features that enterprise application typically use are built in to the EJB framework. You can activate many of these features as necessary, simply by providing the correct annotations. Among those used in the initial version of this course were thread-safety and transaction management. One feature that EJB uses to facilitate thread safety is dependency injection. This ensures that session beans are automatically provided to a servlet – or any client – that needs them in a way that is thread-safe. Our DAO pattern uses a class called the ApplicationContext for this purpose. The application context is a singleton class, and it ensures that all DAO objects used are singletons as well. Our DAO pattern also has an extra level of thread-safety built-in because we decided to make our model classes immutable. This is one
aspect of the DAO pattern that was not discussed in the literature we had available, but the industry professional implemented them this way as a matter of good design.

Transaction management was made explicit in the DAO pattern by implementing the DAO classes using JDBC connections. At the beginning of the transaction, the implementation explicitly had to turn off automatic commitment of newly created data to the database. Once all the data associated with the transaction had been created, a commit of everything was attempted. If the commit succeeded, processing would continue; if it failed, the transaction was rolled back, thereby ensuring that the transaction was atomic. There was certainly more code involved in explicit transaction management, but it was code that was fairly straightforward to debug. In fact, when we introduce transactions to the students, we start by giving them code that fails, and walk through a debugging session with them.

**Knowledge of the latest patterns**

Published descriptions of design patterns typically arise from designs that have been proven in more than one project [1]. That means that an informed industry professional will be aware of designs and techniques before they are widely described in software engineering literature. In our case, the professional used a version of the MVVM pattern (Model View View-Model) to the project that bore only rough resemblance to published accounts.

In the MVVM pattern used, we have one controller servlet for each web page in the application. Each servlet delegates the creation of objects that will be used in the view (a JSP page) to a view-model object. This not only facilitates modularization of the code, but it also allows us to pass a single view-model object to the JSP page. The view-model object contains all the necessary data objects for that page. When we use the jsp:useBean tag to declare the view-model object in the JSP page, it allows our IDE to statically detect all the variables and types in that view-model object. In the first version of the course, students received many warnings from the IDE that a variable or attribute used in a JSP page might not exist. The only fix for this was to declare all such variables using the useBean tag – which made code updates a challenge – or to convince yourself that you had indeed set the attribute in the controller. This practical solution to a limitation of real-world IDEs is not likely to be something found in a research paper.

As mentioned, the MVVM pattern employed bore such little resemblance to the initial MVVM pattern described initially by Microsoft [7] that the educator decided it would be less confusing to students to introduce the pattern by simply demonstrating it in the context of the project and appealing to the benefits described above, rather than have them read any articles regarding the pattern.

These are only a sampling of the modifications made by the profession developer to the Affable Bean project. A more comprehensive account can be found at simpleaffablebean.info. However, we believe that these examples serve to illustrate that many of the changes made – and the way
in which they were made – were not necessarily things you would find in a modern text on web design, or even in a best practices web site for Java web development.

**Sustainability and Student Feedback**

In its present form, the course is moderately sustainable. That is, if a new instructor came in and wanted to use the current course materials, they could do so, but it would take some effort. As the course was designed to be taught online, the video recordings, quizzes, exams, and project descriptions have all been created. The effort for a new instructor would come in understanding the material – particularly the project. Though the project descriptions are fairly straightforward, sometimes the rationale for a design choice is not. The developer’s web site describing his changes to the Affable Bean project would be the best place to start for an understanding of those design decisions, but unless the developer (or the current instructor) were available to answer questions about the project and its design, a new instructor might be advised to go through the project themselves before they taught the course and make any design changes they were comfortable with and could justify to the students. As an example, the way we use the DAO pattern is not documented in any book or online article. A new instructor might feel more comfortable using a version of the DAO pattern that is documented so that they can refer students to specific reading material. It is likely that the previous version of the course was more sustainable because the project closely followed the well-written NetBeans e-commerce tutorial, which instructors could point to for clarification, and students could refer to when they had problems. Unfortunately, the tutorial is significantly out-of-date and many parts of it do not conform to industry best practice.

The student feedback from the first semester of the course – before the developer joined – was generally positive, but there were a few telling comments from students who thought the course was going in the wrong direction. Here is one from a student who thought we were trying to cover too much material in a single semester.

*A lot was missed on both the front end and back end that should be taught. For example, what is event bubbling within JavaScript? This is a very common interview question asked. We barely covered JavaScript within the course. What about common design patterns used for web applications such as Post-Redirect-Get? Web application security was covered in one week. JDBC wasn’t covered at all which in my opinion is more important. What about the DAO or DTO pattern? We used EJB for the course, but Spring is much more common by a large margin.*

It is interesting that once the developer joined the course and redesigned the project, many of this student’s concerns were addressed, even though we had not yet seen the feedback. The revised project uses both the Post-Redirect-Get Pattern and the DAO Pattern, which focuses more on JDBC. It also no longer uses EJB technologies.
Another student was frustrated at the instructor’s lack of ability to adequately respond to questions.

To complete the requirements of this class, I felt like I had to do a lot of additional research outside of the resources provided in this course in order to complete the assignments. Dr. K at times appeared to be confused as he was working through the project assignments at the same time as the class. I feel like the only reason I learned anything in this course beyond the tutorial content was because of the contributions of other students, who were already well versed in web application technologies.

As previously stated, our students had a wide variety of experiences. Some came from a business background and had very little programming experience, while others came from a technical background and had even done some web development professionally. Despite such comments, many students enjoyed the course. Once the developer joined, however, there was a significant increase in overall satisfaction with the course. Figure 3 shows selected results from end-of-semester SPOT surveys for each of the four semesters that the professor has taught the course. The developer joined after the first semester. Statements were given, and students indicated how much they agreed. A score of 1 was given if the student strongly disagreed with the statement, up to a score of 6 if the student strongly agreed. The three questions represented in the diagram for all four semesters are:

- **Teaching Effectiveness**: Overall, the instructor’s teaching was effective. (for Dr. K)
- **Perceived Understanding**: I have a deeper understanding of the subject matter as a result of this course.
- **Stimulated Interest**: My interest in the subject matter was stimulated by this course.

The first statement pertains only to the professor (Dr. K) as only he had survey results both before and after the developer joined. The other two statements concern the course in general. It is evident from the chart that the introduction of the developer to the course significantly raised all three scores. On average, students were more likely to feel that they had a deeper understanding of the subject matter after the developer joined the course. Likewise, they were more likely to say that their interest in the subject matter had been stimulated. Finally, they were even more likely to feel that the professor was doing an effective job. This last part is particularly interesting because the first semester the developer was involved was a bit chaotic as we were trying to quickly modify the project (and therefore the other materials as well) as the semester progressed, and the professor certainly fell behind in some of his duties. The line in the diagram with only three data points pertains to the developer. The statement given was: “The instructor related theories and concepts to practical issues.” We expected this statement to receive significant agreement from students, and it did.
The developer received glowing written reviews from many of the students. We list a few of their comments below.

*Dr. A provided clear explanation of why our project code executes and behaves the way it does. He also provided real world insight into the purpose of the code we developed.*

*[Dr. Atkinson] was incredibly valuable to the course, and I learned a great deal from his insight, comments and feedback during both the Q&A lectures as well as the Piazza forum.*

*Dr. Atkinson brings many years of professional experience. His field experience added bonus value to the class. He has been very helpful on the discussion board throughout the semester, and I enjoyed his debugging sessions. I am glad that Dr. A was in my class, his advice and help have been priceless.*

There are many, many such comments. But perhaps our favorite comment with regard to Dr. Atkinson is this one:
One of the things everyone mentions these days is to learn from experts and you taught me a great deal just by how you approached items, were willing to provide out of the box problem solving or look at an issue, and I liked having a non-academic (and I do not refer to your credentials, it is just you have such great practical experience) have some input. I love that you and Dr. K can work on something, disagree at times, but have such great mutual respect.

Challenges ahead and conclusion

In general, we have had a very positive experience in collaborating to teach our web application development course. As with any experience, there were areas that needed to be improved. There were times when coordination could have been better. Sometimes the educator would go into a Q&A session trying to accomplish a certain programming goal and the professional had problems with how the code was structured. Usually these problems were small enough to where we could work things out within the session and it would turn into a good “learning opportunity” for the students. But sometimes we needed to discuss things after the session and announce a correction or explanation of the right way to do things. Sometimes portions of projects were a little ambitious. The professional might be excited to introduce a new design or technique that was challenging to many students. If the educator caught this, he could correct it; if he was swept up in the enthusiasm as well, the project would have to be dialed back after the fact to something more manageable. At the beginning of the collaboration, perhaps the biggest problem was the time that both the professional and educator had to put into the course. The professional was anxious to make all the changes to the project quickly. But of course, the code can always be improved, so he needed to decide when those changes were sufficient. Meanwhile, the educator needed to make his own changes to the course – the readings, the slides, the lectures – to reflect the changes the professional was making. Though we continue to make changes to the course every time we teach it, the major up-front costs in terms of time seemed to have settled down.

We know that it can be difficult to take what happens in a single course and try to apply it to a course with a different student make-up, a different learning environment, and a different set of projects. This is true even when you have the same kinds of students, the same learning environment, and even the same collaboration, and you just want to create a different development course. We know because that is what we are currently trying to do for a Mobile Application Development course.

The following points summarize the broad sections of this paper.

1. For a collaboration to be successful, academic goals and objectives must take priority. No amount of professional knowledge about tools and techniques can change this.
2. Both the educator and the industry professional excel at different things in the learning environment. Leverage these different abilities for the benefit of the students.
3. Utilize modern distance learning tools like discussion boards and online Q&A sessions even if you are not teaching an online course. They will make your life easier and can actually facilitate interaction between students and teachers.

4. Leverage the professional’s development experience when you need to develop tools that make the course easier to manage and the projects easier to grade.

5. Making course projects meaningful and ensuring well-designed code is where professional developers really shine. The educator should listen to that expert knowledge but temper it with an understanding of what is best for the students.

In the Mobile Application Development course that we are currently teaching (for the first time), we tried to look at what worked and what needed to be improved from our collaboration in the Web Application Development course. We find that most of the challenges are coming from the fact that it is a new course. Therefore, we are still moving lessons and projects around. Even so, a few incidents have already occurred that proved the benefit of having a professional developer help with the course. For one of the Q&A sessions, the educator prepared a sample project that would help students get a start on the project they had to turn in. The educator shared the description of the sample project with the professional and implemented it so that it functioned correctly. During the presentation, however, the professional noticed that the implementation partly conflated the Model and View portions of the Model-View-Control pattern. After the Q&A and a brief discussion, we decided to hold another Q&A session in which the professional refactored the educator’s code to eliminate this problem. This sparked a conversation in the online discussion forum about the Model-View-Controller pattern as it applied to Android development: How does Android use the MVC? Is it a true MVC pattern? Are there multiple versions of the MVC? The response from the developer answered the questions in a way that a response from the educator could not. We end this paper with the response and present it as another example of why educators should team with industry professionals in software development courses.

In practice, what I’ve seen is that indeed there are many variants of MVC. Some of that depends on the framework, but some of it also depends where the “network” lives and what sort of system we are building. In an Android system, we are talking about organizing code that all runs together on a device. In a website architecture, we are talking about (classically) a browser being a view, with the model and controller being on the server (as we taught in web application development). In a client-side web architecture, you can have the controller and view in the browser, with the model being on the server (think Angular, Vue, React etc). So, I’d say it’s important to know what you are building, and which choices are available. It seems like Activities / Fragments / View Adapters / Models / Layouts and View objects is the way the Android world has split things up. An important takeaway from this is to know when to “fit in” to which patterns, when to work around them, and how to keep your code small and clean no matter which architecture you decide upon.
References

[1] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, 1994.


