

# Studios, Mini-lectures, Project Presentations, Class Blog and Wiki: A New Approach to Teaching Web Technologies

**Xuesong (Sonya) Zhang**  
**California State University**  
**Fresno, CA, USA**

[xzhang@csufresno.edu](mailto:xzhang@csufresno.edu)

**Lorne Olfman**  
**Claremont Graduate University,**  
**Claremont, CA, USA**

[lorneolfman@cgu.edu](mailto:lorneolfman@cgu.edu)

## Executive Summary

Designing web technology courses that match students' knowledge background and emphasize students' hands-on experience is imperative yet challenging. This paper discusses a new approach to teaching web technologies, particularly, open source Web 2.0 technologies, in which a combination of studios, mini-lectures, presentations and tutorials, class blogs, and wikis is used. We reported two case studies to demonstrate and evaluate how this approach was used in two web technology courses - one graduate level and the other undergraduate. Based upon survey results and written evaluations from the class blogs and wikis, students feel this new approach makes the course one of their most positive and valuable learning experiences. Educators teach various new and advancing web technologies in Information and Computer Science courses today. Tested and evaluated in Web 2.0 technology courses, we believe that this studio centered approach can benefit both teaching and learning in a variety of web technology courses.

**Keywords:** web technologies, teaching, learning, open source, Web 2.0.

## Introduction

Today's World Wide Web offers dynamic services and applications that emphasize interactive information sharing, interoperability, user-centered design, and online collaboration. These technologies embody the philosophy of a user-centered Web, referred to as Web 2.0 (O'Reilly, 2005). Meanwhile, our students are keen on using technologies in their social life - the Internet, email, instant/text messaging, computer games, and, more recently, social networks such as Flickr, YouTube, MySpace, Facebook, Twitter, and Digg. Students' familiarity with digital technologies and the emergence of Web 2.0 facilitate the application of social constructivist pedagogy in higher education. Teaching Web Technologies has become a more imperative yet challenging task in this changing environment, particularly the task of how to design courses that match the back-

---

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact [Publisher@InformingScience.org](mailto:Publisher@InformingScience.org) to request redistribution permission.

ground and learning style of new generation students, and foster both knowledge and practical skills.

Both objectivism and constructivism contributed to computer science education, software engineering education, and e-learning (Kamthan, 2009). In the objectivist view, knowledge is considered to be external to an individual, and learning involves a transfer of knowl-

edge from the instructor to the student. Constructivist learning theories (Bruner, 1966; Piaget, 1950; Wood, 1998; Vygotsky, 1978) on the other hand suggest that learning is an active, social process; learners develop or construct their knowledge more effectively through their own experiences and interaction with others than through traditional objectivist methods, such as lectures and exams. Examples of active learning processes promoted by constructivist learning theory include hands-on experiments, lab assignments, group projects, and field studies. Constructivist learning theories also suggest that collaborative learning is critical in constructive cognitive development (Piaget, 1928). Collaborative learning brings students to work and learn together. In collaborative learning, interaction between peers (students) is equally shared, thereby helping sustain the students' interests and providing more natural learning environment. Both constructivist and collaborative learning theories suggest that a teaching approach that promotes more student hands-on practice, interactive information sharing and collaboration would benefit student learning more than traditional lecture and exam based course design.

## A Studio Centered Approach to Teaching Web Technologies

In the following, we describe the detailed design (including class activities, objectives, description, suggested usage, and instructor's and students' role) of a studio centered approach to teaching web technologies.

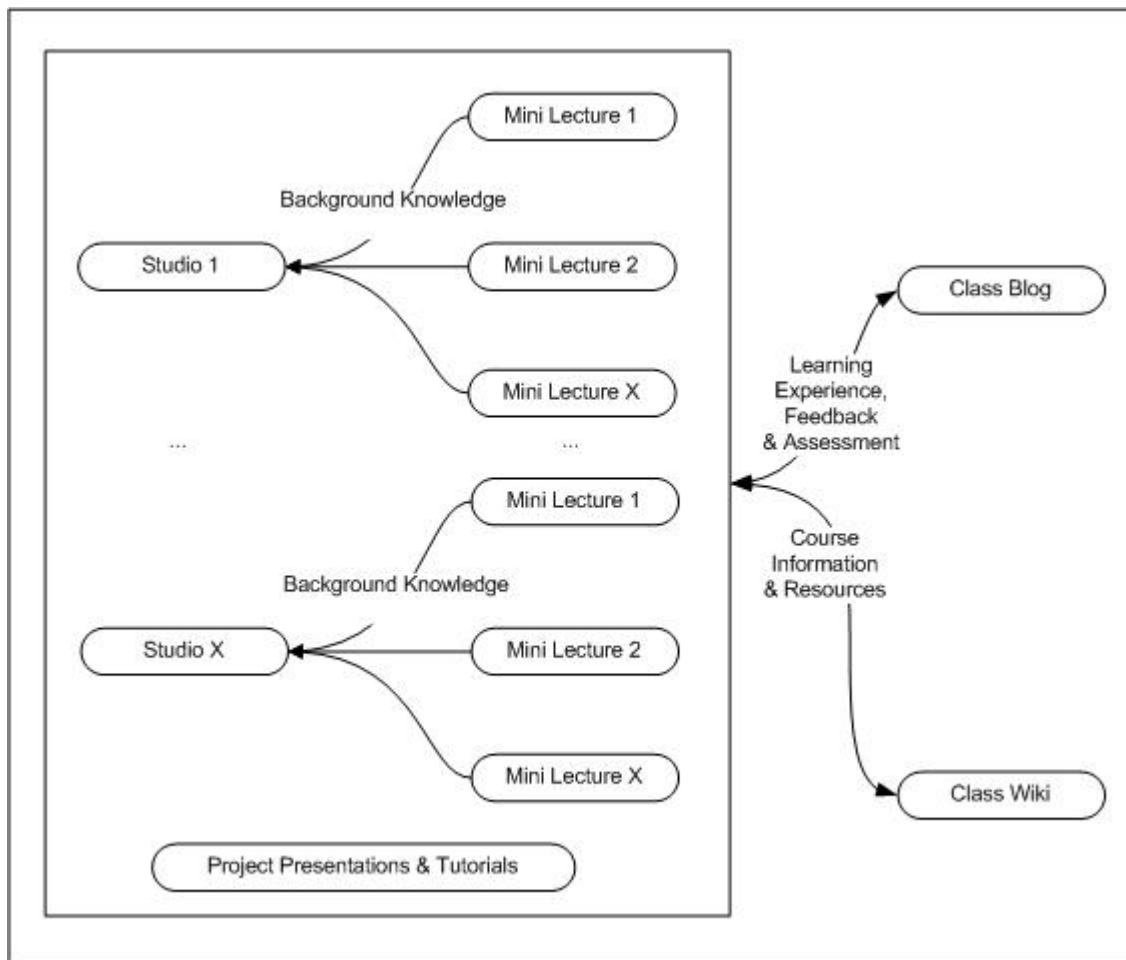


Figure 1: A Graphical View of Class Activities

## ***A Studio Centered Approach***

Based on the above discussion, we propose a new approach to teaching web technologies, which is studio-centered, supplemented with mini-lectures, project presentations and workshops, and class blogs and wikis. Figure 1 shows a graphic view of the organization and interaction of various class activities within this approach. Each Studio is supported by a number of mini lectures that explain the relevant knowledge concepts and industrial examples. The project presentations and tutorials showcase complex web applications and web-based systems that provide real business solutions. The class blog and wiki provide communication channels for students and instructor(s) to share information, resources, and feedback.

Table 1 shows the types of class activities, including objectives, a brief description, and suggested usage in this new approach.

**Table 1: A Studio Centered Approach to Teaching Web Technologies**

<b>Types of Class Activities</b>	<b>Objectives</b>	<b>Description</b>	<b>Suggested Usage</b>
<b>Studios</b>	Enhance student learning through hands-on practice, interaction and collaboration.	Students install, configure, customize and manage various Web technologies/software instances in a virtual workspace such as the LAMP (Linux, Apache, MySQL, and PHP) environment.	Yes
<b>Mini-lectures</b>	Provide background knowledge that relates to the specific web technologies covered by the studio.	Instructor presents the basic knowledge and concepts that relates to web technologies, examples are: web hosting, programming, open source software, system development life cycle, etc., depending on the studio topics.	Yes
<b>Presentations and Tutorials</b>	Introduce other web technologies/projects that are not covered by the studios, but are up-to-date and valuable in practice.	Instructor or guest speaker introduces other web technologies/projects that are not covered by the studios. It can be in the form of a presentation, demo, and/or tutorial in class.	Optional
<b>Class blog</b>	Document and share students' learning experiences; obtain feedback and assessment.	Students document and share their learning experience in the class. Instructor and classmates provide their feedback and suggestions by commenting on the blog entries. The class blog entries can also be used as an assessment element in grading.	Optional (Classroom assessment tools such as survey, email, or discussion forum can replace it.)
<b>Class Wiki</b>	Publish and share course related information and resources.	Instructor publishes Syllabus, Studio Instructions, Grading and Expectations, Student Web Site URLs, MediaWiki Help, and Exercises. Students contribute to Group lists, Group Project Topics, Group Project presentations, and their project home page URL (i.e., index page). The project home page consists of links to the student's various projects (WordPress, Mediawiki, phpBB, mashups) and a short description of each project.	Optional (Course Management Systems such as Blackboard, WebCT, Sakai, or Moodle can replace it.)

### ***The Instructor's Role***

In addition to being a leader and administrator as in traditional classrooms, the instructor in this new approach also acts as a designer and facilitator, similar to the roles of instructors for online classes. The instructor designs the contents and arrangement of the studios and mini lectures, invites guest speakers for presentations and tutorials, monitors class blogs and wikis, and facilitates various classroom activities to ensure effective and efficient learning outcomes. These new roles require stronger instructor support, more frequent instructor-student interaction, and superior organizational skills than being an instructor in the traditional lecture and exam based environment.

### ***The Students' Role***

As the new approach helps create an environment of active, engaged, and exploratory learning, students become more independent and responsible learners. Students develop higher-level critical thinking, problem solving, communication, and social interaction skills compared to traditional lecture and exam based courses. In addition, some students could also develop leadership and management skills during the process of helping and/or collaborating with classmates.

## **The Case Studies**

### ***The Class Environment***

We used the aforementioned studio approach in a graduate class (Fall 2008) and an undergrad class (Spring 2009) in two different U.S. universities. The graduate class consisted of 12 students majoring in Information Systems and Technology. The undergrad class consisted of 21 students majoring in Computer Information Systems in a business school. All students had taken fundamental computer literacy classes as prerequisites. The course was named “Social Technologies” in the graduate class and “Open Source Web 2.0 Technologies” in the undergrad class. Both classes covered the same materials, following the same approach as demonstrated in Figure 1 and Table 1.

### ***Learning Goals and Objectives***

The goal of the class is to learn the range of open source Web 2.0 technologies (including blogs, wikis, discussion boards, and mashups) that are available, the range of capabilities they possess, how they can be used and evaluated effectively, how they can be modified to enhance their usability, and how to design systems that use open source Web 2.0 technologies to support effective Knowledge Management. The learning objectives for the courses are as follows:

- 1) Install and configure various Web 2.0 software instances (such as WordPress, MediaWiki and phpBB2) in a LAMP (Linux, Apache, MySQL, and PHP) environment.
- 2) Download, install and configure various plug-ins and add-ons for Web 2.0 software instances.
- 3) Establish and maintain user accounts for Web 2.0 software instances.
- 4) Customize instances of Web 2.0 software within a knowledge management context.
- 5) Install, configure, customize, and maintain an instance of Web 2.0 software other than the ones used in class exercises and demonstrations.
- 6) Understand and explain basic concepts such as Web hosting, open source software, Web 2.0, and knowledge management.

## Class Activities

A student background survey was done at the beginning of the class to understand the students' IT skill levels and experiences using Web 2.0 technologies. The survey results helped the instructors adjust the design of the studios (e.g., the difficulty level of tasks, the detail level of instructions, and the grading criteria) and select the appropriate topics of mini lectures to match with students' background and needs.

The instructors have designed the course slightly differently between the graduate and undergrad class, based on the assumptions that graduate students generally have more relevant IT experience and background and also have taken more advanced pre-requisites before this course. The differences are described in the following:

- More details were provided in the studio instructions and mini lectures, and grading criteria and expectations were lowered for the undergrad class.
- The student project was an individual project in the graduate class and a group project in the undergraduate class.
- The graduate class focused on using the web technologies as tools for knowledge management, which means that the knowledge content must be meaningful, organized, and structured. The undergrad class focused on the web technologies (e.g., installation, configuration, customization, etc.), which means that knowledge content can be of lower quality or informal (e.g., copy and paste from other web sites).
- The instructors used examples closer to the undergrad students' everyday life in class, for example, a mini lecture was given to the undergrad students about Facebook and MySpace to help them understand the critical success factors for running social networking websites and to stimulate their interests in learning the underlying technology, because most of them have used both. In the graduate class, the instructor used examples closer to the corporate environment and also encouraged students to share their relevant work experience.

The detailed class activities for the courses were designed based on the studio approach (see Table 2).

**Table 2: Detailed Class Activities**

Topics	# of class sessions	Description	Mini Lectures
Setting up Virtual Workspaces	1	<ul style="list-style-type: none"> <li>• Get familiar with file management and MySQL database settings on the Web hosting site.</li> <li>• Download and install FireFTP or other FTP clients</li> <li>• Create project home page (i.e., index page) ready for publishing project links and descriptions.</li> </ul>	<ul style="list-style-type: none"> <li>• Web 2.0</li> <li>• Open source software</li> <li>• Web hosting</li> <li>• LAMP &amp; WAMP</li> <li>• Knowledge management</li> </ul>
Web blogs	2.5	<ul style="list-style-type: none"> <li>• Download and install WordPress.</li> <li>• Login as admin, create and manage posts, pages, comments and categories; install new themes; customize settings.</li> <li>• Comment on classmates' blog posts.</li> <li>• Download, install, and manage WordPress plugins &amp; widgets</li> </ul>	<ul style="list-style-type: none"> <li>• Blogs</li> <li>• WordPress</li> <li>• PHP programming</li> <li>• Application Programming Interface (API)</li> <li>• Integrated Development Environment (IDE)</li> </ul>

Topics	# of class sessions	Description	Mini Lectures
<b>Wikis</b>	2.5	<ul style="list-style-type: none"> <li>Download and install MediaWiki.</li> <li>Login as admin, configure and customize logo; install new skins; create new pages, links, categories, and templates; customize settings.</li> <li>Contribute to classmates' MediaWiki pages.</li> <li>Download, install, and manage MediaWiki extensions.</li> </ul>	<ul style="list-style-type: none"> <li>Wikis</li> <li>MediaWiki</li> <li>MySpace vs. Facebook</li> </ul>
<b>Discussion Forums</b>	2	<ul style="list-style-type: none"> <li>Download and install phpBB.</li> <li>Login as admin, install new style, customize logo, create forums, sub-forums, and topics under forums.</li> <li>Respond to classmates' forum topics.</li> <li>Install phpBB Mods.</li> </ul>	<ul style="list-style-type: none"> <li>Discussion Forums</li> <li>phpBB</li> <li>phpMyAdmin</li> </ul>
<b>Mashups</b>	1	<ul style="list-style-type: none"> <li>Create two working mashups using Microsoft Popfly (discontinued on August 24, 2009).</li> <li>Make these available on student home page (i.e., index page).</li> </ul>	<ul style="list-style-type: none"> <li>AJAX</li> <li>Mashups</li> <li>Google Maps</li> <li>Microsoft PopFly and Microsoft Silverlight</li> <li>Yahoo Pipes</li> <li>Google Mashup Editor</li> </ul>
<b>Other Web 2.0 Projects</b>	3	<ul style="list-style-type: none"> <li>KEEP SLS – an open source, social-learning based ePortfolio system.</li> <li>Claremont Conversations Online – a social networking-based virtual community for higher education based on Elgg.</li> <li>Multimedia Management and Presentation System (MMPS) – an interactive and collaborative multimedia management and presentation system.</li> </ul>	

## Studios

The studios feature hands-on in-class activities that enable students to share their learning immediately and to ask for assistance in understanding how specific Web 2.0 technologies work in practice.

Each student was given an individual web and database workspace set up in a LAMP (Linux, Apache, MySQL and PHP) environment. During class sessions students learned how to install, configure, customize, and manage various Web 2.0 software instances. This process was facilitated through a set of exercises that students did during class and for homework. The instructor posted instructions for each studio on the class wiki. Students documented their learning using the class blog. Students were encouraged to share their findings with other classmates and ask classmates and instructors for help in doing activities. Students were also encouraged to comment on blog entries posted by other students.

There were a total of 12 studio sessions (see details in Table 2), which covered setting up the virtual work space and four open source Web 2.0 technologies: WordPress blog, MediaWiki, phpBB, and Microsoft Popfly mashup editor (discontinued on August 24, 2009). In addition, some up-to-date Web 2.0 projects were introduced to the students in the form of presentations and tutorials. These projects included an open source ePortfolio system, a virtual community, and a multimedia management system.

## **Mini lectures**

The instructors gave several mini lectures (about 20-30 min in length) at the beginning of each class session, explaining the background knowledge that relates to the specific technologies the students would be working on that day; for example, Web 2.0, LAMP, Routing, Web hosting, and knowledge management were discussed in the first studio, when students set up their virtual work spaces.

## **Student project**

The purpose of the student project is to enable students to become an expert on at least one additional open source Web 2.0 technology that was not covered during class sessions. In gaining this expertise, students demonstrated progress toward the goal by doing research to identify the Web 2.0 technology chosen, presenting their preliminary work during a class presentation, installing and customizing their Web 2.0 technology instance to enhance knowledge management activities in a context they chose, keeping a record of the process of becoming an expert through class blog entries, and presenting their final project on the last day of class. Students chose various Web 2.0 technologies as their projects, including content management systems such as Drupal, Joomla, Mambo, and Moodle; ecommerce solutions such as osCommerce and Magental; galleries such as Coopermine and Gallery; and social networking systems such as Elgg.

## **Class web blog**

Students used a class web blog to document and share their learning experiences. The instructor and other students provided feedback and suggestions by commenting on the blog entries. A certain number of class blog entries were required for each Web 2.0 technology.

## **Class wikis**

The instructor used a class wiki site to publish course related information and resources such as the Syllabus, Studio Instructions, Grading and Expectations, Student Web Sites URLs, MediaWiki Help, and Exercises. Each studio instruction set contained step-by-step in-class exercise instructions and homework. Using the collaborative writing feature that wikis provide, students also contributed some content on the class wiki including Group lists, Group Project Topics, Group Project Presentations, and Group Project Site URLs.

## ***Grading and Expectations***

There were a total of 12 studios, and a maximum of 350 points could be earned for the studio assignments. Studio activities were mainly graded based on the complexity of the technology (e.g., WordPress and MediaWiki were more complicated than phpBB in terms of customization), functionality, and completeness. General technical aspects considered in grading were look and feel, plug-ins or add-ons, contents, and class blog posts. The project was worth a maximum of 250 points. Grades were based on progress toward acquisition of expertise, recording of progress via blog entries, and the final project presentation. The instructor published detailed rubrics on the class wiki site so the students could have a clear understanding of the grading criteria and expectations. Table 3 shows detailed grading and expectations.

**Table 3: Grading and Expectations**

Topics	Grading and Expectations
<p><b>WordPress (100 points)</b></p>	<ul style="list-style-type: none"> <li>• <b>Look and Feel</b> of the blog (30 points)                             <ul style="list-style-type: none"> <li>◦ Install a working and pleasing theme, reorganize the sidebar, and so forth.</li> <li>◦ Create at least 4 blog posts, 1 blog page, a few tags, and categories.</li> </ul> </li> <li>• <b>Plugins</b> (30 points)                             <ul style="list-style-type: none"> <li>◦ Install at least 5 plugins and make sure they work. Students were expected to explain what they are and how they work in the class blog site. In addition to the 5 plugins, register at WordPress.org and get an Akismet key so the Akismet (comes as a default WP plugin) works properly.</li> </ul> </li> <li>• <b>Contents</b> (20 points)                             <ul style="list-style-type: none"> <li>◦ Create the contents of the blog site as an instance of a KM system in an organization, including meaningful tags and categories as well as content that matches a theme.</li> </ul> </li> <li>• <b>Comments</b> on classmates' blogs (10 points)                             <ul style="list-style-type: none"> <li>◦ Earn 1 point for commenting on each classmate's blog site, up to a maximum of 10.</li> </ul> </li> <li>• <b>Class Blog Posts</b> (10 points)                             <ul style="list-style-type: none"> <li>◦ Earn 10 points for 2 or more posts (2=10, 1=5, 0=0).</li> <li>◦ Create at least two blog posts on the class blog site: one is about blog technology in general, and the other is about the experience of installing the plugins.</li> </ul> </li> </ul>
<p><b>Wiki (100 points)</b></p>	<ul style="list-style-type: none"> <li>• <b>Look and Feel</b> of the wiki (40 points)                             <ul style="list-style-type: none"> <li>◦ Install a custom logo, add categories, links/new pages, and templates, and try different skins.</li> <li>◦ Create at least 4 new pages (show links on the wiki main page so the instructor can easily locate them), 2 links (one external link such as www.cnn.com, and one email link), 2 categories, and 2 templates.</li> <li>◦ Use one of the 8 default MediaWiki skins (because they usually work well with many extensions, and a customized logo).</li> <li>◦ Add the instructor as a user to the MediaWiki site and grant administrator permission so the instructor can log in and check the skins installed.</li> </ul> </li> <li>• <b>Extensions</b> (40 points)                             <ul style="list-style-type: none"> <li>• Install at least 5 working MediaWiki extensions, including the Ajax show editor extension demonstrated in the class.</li> </ul> </li> <li>• <b>Content</b> (10 points)                             <ul style="list-style-type: none"> <li>◦ Create the contents of the MediaWiki site as an instance of a KM system in an organization, including meaningful tags and categories as well as content that matches a theme.</li> </ul> </li> <li>• <b>Class Blog Posts</b> (10 points)                             <ul style="list-style-type: none"> <li>◦ Earn 10 points for 2 or more posts (2=10, 1=5, 0=0).</li> <li>◦ Create at least two blog posts on the class blog site: one is about wiki technology in general, and the other is about the experience of installing MediaWiki extensions.</li> </ul> </li> </ul>



Topics	Grading and Expectations
<b>phpBB (100 points)</b>	<ul style="list-style-type: none"> <li>• <b>Look and Feel</b> of the forum (25 points) <ul style="list-style-type: none"> <li>◦ Install a custom style, a logo, and announcements located on top of the forum site or somewhere obvious to see.</li> </ul> </li> <li>• <b>MODs</b> (40 points) <ul style="list-style-type: none"> <li>◦ Install at least 5 working phpBB MODs, in addition to the ACP announcement MOD, which was demonstrated in class.</li> </ul> </li> <li>• <b>Content</b> (25 points) <ul style="list-style-type: none"> <li>◦ The instructor rated content on a scale of 0-5, then allocated points as follows: 5=25, 4=20, 3=15, 2=10, 1=5, 0=0.</li> <li>◦ Contents were based on: <ul style="list-style-type: none"> <li>◦ Create the contents of the phpBB forum as an instance of a KM system in an organization, including meaningful tags and categories as well as content that matches a theme.</li> <li>◦ Create at least 3 forums that have different content related to the KM theme, with 1 subforum in one of the forums; at least 3 topics in each forum; and at least 2 replies to each topic, in addition to the replies the classmates made.</li> <li>◦ Post at least 2 announcements via the ACP announcement MOD.</li> <li>◦ Post replies to at least 10 of the classmates' forums.</li> </ul> </li> </ul> </li> <li>• <b>Class Blog Posts</b> (10 points) <ul style="list-style-type: none"> <li>◦ 2 or more posts earn 10 points, 2=10, 1=5, 0=0.</li> <li>◦ Create at least two blog posts on the class blog site: one is about online forums and/or phpBB in general, and the other is about the experience of installing phpBB MODs.</li> </ul> </li> </ul>
<b>Mashups (50 points)</b>	<ul style="list-style-type: none"> <li>• <b>2 working mashups</b> using Microsoft Popfly (30 points) <ul style="list-style-type: none"> <li>◦ Each working mashup earns 15 points, 2=30, 1=15, 0=0.</li> <li>◦ Make these mashups available on the student's project home page (i.e., index page)</li> </ul> </li> <li>• <b>Class Blog Posts</b> (10 points). <ul style="list-style-type: none"> <li>◦ Create at least one class blog entry to discuss knowledge and experience about mashups. Also in this blog post, explain how the two installed mashups work.</li> </ul> </li> </ul>
<b>Student Projects (250 points)</b>	<ul style="list-style-type: none"> <li>• <b>Actual project (200 points)</b> <ul style="list-style-type: none"> <li>◦ Install only one instance of the chosen software on the class web space.</li> <li>◦ Look and feel (70 points)</li> <li>◦ Customization (80 points)</li> <li>◦ Contents (50 points)</li> </ul> </li> <li>• <b>Presentations (50 points)</b> <ul style="list-style-type: none"> <li>◦ PowerPoint slides showing the history and background of the software you chose, a demo of the software, and a conclusion (what is learned from the experience, what is the value of the software, what are future perspectives of the software, etc.)</li> <li>◦ Demonstrate the admin panel, main features in this web application, also customization (find and install at least a different skin/theme, and a couple of modifications/extensions/plugins that didn't come with the default installation), and knowledge contents.</li> </ul> </li> </ul>

## Survey

A survey questionnaire was designed to find out the students' perspective on the effectiveness of this studio approach. The survey was given to the undergrad class at the end of the term; 21 valid questionnaires were collected.

Table 4 shows the results. All students (12 strongly agree, 9 agree) indicated that the software was easy to install and use. Students also agreed that because of the studio approach, they improved their skills in using information technologies and self-learning, they learned more about open source software and Web 2.0 technologies than before, and they were more motivated in the class. The majority of the students said they learned the most from customizing the software (e.g., adding themes, plugins, extensions, and modifications), debugging and problem solving, and research online or via other resources outside the classroom to learn how to install, configure, and/or customize the software.

Among all the questions in the survey, the one that received the most neutral responses (2 out of 21) was “Sharing, communicating (comments, feedback, suggestions, etc.), and collaborating with classmates.” All students (19 strongly agree, 2 agree) preferred the new course design to traditional lectures for this subject. With the same distribution, students thought it would be a good idea to have the new design in other information or computer science courses. It is implied that students’ positive experience from this particular course influenced their expectations and interests towards other information systems and technology courses.

**Table 4: Survey Results**

\*SD = Strongly Disagree, D = Disagree, N = Neutral, A= Agree, SA = Strongly Agree

Item	SD*	D*	N*	A*	SA*	Total
The software introduced in this class is easy to install and use.	0	0	0	9	12	21
Because of the studio approach,	0	0	0	7	14	21
- I improved my skills in information technologies.	0	0	0	5	16	21
- I improved my self-learning skills.	0	0	1	5	15	21
- I improved my problem solving skills.	0	0	0	5	16	21
- I learned more about open source software.	0	0	0	3	18	21
- I learned more about Web 2.0 technologies.	0	0	0	5	16	21
- I am more motivated.	0	0	0	5	16	21
- I learned a lot in this course.	0	0	0	5	16	21
I learned a lot from:	0	0	1	9	11	21
- Installing the software.	0	0	1	6	14	21
- Configuring the software.	0	0	1	3	17	21
- Customizing the software (adding themes, plugins, extensions, and modifications)	0	0	1	4	16	21
- Debugging and problem solving.	0	0	1	4	16	21
- Research online or via other resources outside classroom to learn how to install, configure and/or customize the software.	0	0	1	4	16	21
- Sharing, communicating (comments, feedback, suggestions, etc.), and collaborating with classmates.	0	0	2	4	15	21
I prefer the new studio approach (a combination of studios, mini-lectures, presentations and tutorials, class blogs and wikis) to traditional lectures for this subject.	0	0	0	2	19	21
It would be a good idea to have new design in other information science or computer science courses.	0	0	0	2	19	21

## ***Student Feedback from the Class Blogs and Wikis***

Students were asked to make final comments about the class at the end of the term. Graduate students posted such comments on a virtual community site as a wiki page, while undergrad students posted such comments on the class blog site. The comments were very positive. Even though in the beginning, many students were concerned that the class would be as difficult as many other programming classes, their opinions quickly changed as the class progressed. Students commented on how the studio approach helped enhance their learning experience. They enjoyed both the studios and mini lectures and helping each other in the class. They also thought building production web sites via a live web server was motivating and rewarding. They indicated increased confidence and interest in information systems and technology courses and degrees after taking this course. Some examples of comments follow:

My learning experience in this class has been great. This was my first time installing and using web 2.0 apps in a hands-on approach. I think that the lab-class setting has great influence in keeping most students motivated with the studios. The mini-lectures were also important, especially to understand what's going in the background of every application that we used.

I had previously a few other classes that required building and uploading a site to a live server, but it wasn't as interactive as this class was. Most apps were really quick and easy to install, and I now know that in less than an hour I can have WordPress, MediaWiki, phpBB and osCommerce with some mashups on all of them to enhance the visitor's experience. The overall experience was excellent and I know that I will use this experience in the near future.

Overall, my experience with web 2.0 was a great one. This was one of the best classes I have ever had in [school name removed] because it allowed me to work on hands on projects instead of just listening to lectures. I love that this class did not stress me out by having tests and quizzes. I loved the WordPress project because I found it to [be] the easiest, yet creative. I learned a lot about plugins, and modifications. One of the most challenging parts of this course was phpbb3 because I remembered spending a lot of time on all the modifications. Another interesting project was the mash up because it allows me to create and add features very easily. I am so glad I took this class after debating whether the class is going to be too difficult or too easy. This new knowledge of web 2.0 technologies is great to have and I hope to use it more in the near future.

The experiences that I have taken from this class have given me the opportunity to learn about the many new features of what Web 2.0 has to offer. Before taking this class I knew very little about web 2.0 and the technologies involved, and when I saw that this course was being offered I knew I had to jump in and take it. I did not know what to expect from it, but from the very first studio when [the instructor] described the workshops were going to be hands on and interactive — I knew I was going to enjoy this course because I really enjoy that type of teaching style.

The hands-on approach to learning about web 2.0 technologies has kept me engaged in every studio lecture making it that much more enjoyable. Being able to work on the projects in class gave me plenty of time to work things out and if I ever encountered a problem, I had my classmates or [the instructor] around to help me out. Because of this - the open atmosphere of students helping each other out made getting the work done simpler and enjoyable.

This class has been one of the few fun classes that I have taken. The workload is balanced even and the studios are engaging and fun. I also really enjoyed learning about what the latest web 2.0 trends are and [it is] a source of keeping me up to date. This has been a very enjoyable quarter and if this class is to be taught again, my suggestion would be to keep the teaching style how it is!

### ***Challenges Faced During the Course***

As in any other web technology course, the instructor needs to help students of all skill levels tackle and overcome their technology challenges. Some students needed extra time and help to work on a studio but the instructor could only spend limited time with each student during class sessions. It was often a challenge to balance the class, without making the students who were slow-paced feel frustrated or left out or making the others feel bored waiting for them to catch up.

One solution the instructors applied was encouraging students to help each other. Another solution was designing the studio with clear and straight forward instructions. That way even though the student couldn't complete the studio in class, he or she could finish on their own after class following the instructions. Another solution was sharing learning experiences on the class blog. Students were asked to blog about their lessons learned for each studio. This helped extend learning even if the student did not encounter the same problem as others did or did not have a chance to interact or collaborate with others in class.

Both classes used external web hosting services to host student projects, class blogs, and class wikis. The benefit of choosing a web hosting service provider is that the web server (often Apache), database (often MySQL) and programming language (often PHP or Perl) environment is pre-configured and ready to use. They also provide user support, maintenance, and backups. The most important advantage is that, with the public URL and FTP access to the files on the remote web hosting site, it is easier for students to share projects with the class and to collaborate on group projects. However, this approach also presented problems. During the class session, it was not uncommon that over 20 students were simultaneously transferring files by FTP to the remote web hosting site, which could significantly slow down the traffic and even drop connections at times. Another problem was that many students tended to work on the assignments and projects late at night, which was also the common time slot for the web hosting services to perform maintenance and backup. Therefore their connection and speed were often compromised.

As an alternative, the instructor can choose to use a web server that is either co-located at an Internet Service Provider or on a campus network. The students can still share and collaborate via FTP as long as a domain name or public IP address and FTP accounts are provided. The speed can be dedicated and customized toward the demand of the course. This alternative is better suited for instructors who have some level of IT experience and do not mind the overhead involved in installing and managing server software and user accounts by themselves.

### **Conclusion**

How can instructors teach emerging web technologies to students through adequate hands on practice within a short period of a time? How can students benefit from learning from each other rather than solely relying on the instructor's lecture and lab instructions? The survey results and students' evaluations of this study indicate that a studio centered approach to teaching web technologies, combined with mini-lectures, presentations and tutorials, class blogs and wikis, may provide a solution to creating active and collaborative learning while fostering skill building and practice. Such a design may also be a valid alternative to more traditional course design in the subject.

Educators teach various new and advancing web technologies in Information and Computer Science courses today. Typical examples include: LAMP (Linux, Apache, MySQL, and PHP/Perl) or WAMP (Windows, Apache, MySQL, and PHP/Perl), Python, Ruby on Rails, Microsoft .Net, Java Enterprise and SOA (Service Oriented Architecture). Although the case studies in this paper only demonstrated how the proposed studio approach was applied in classes that taught Web 2.0 technologies, we encourage other researchers and teachers to test and evaluate using the studio centered approach in a variety of web technology courses.

## References

- Bruner, J. S. (1966). *Toward a theory of instruction*. Cambridge, MA: Harvard University Press.
- Kamthan, P. (2009). A methodology for integrating the social web environment in software engineering education. *International Journal of Information and Communication Technology Education*, 5(2), 21-35.
- O'Reilly, T. (2005). *What is Web 2.0*. Retrieved June 28, 2010, from: <http://oreilly.com/web2/archive/what-is-web-20.html>
- Piaget, J. (1928). *Judgement and reasoning in the child*. New York: Harcourt Brace.
- Piaget, J. (1950). *The psychology of intelligence*. New York: Routledge.
- Wood, D. (1998). *How children think and learn* (2nd ed.). Oxford: Blackwell Publishers Ltd.
- Vygotsky, L. S. (1978). *Mind and society: The development of higher mental processes*. Cambridge, MA: Harvard University Press.

## Biographies



**Sonya Zhang** is an assistant professor in the Craig School of Business at California State University Fresno. She received a PhD in Information Systems and Technology from Claremont Graduate University, a Master of Science in Applied Computer Science, and a Master of Business Administration from Illinois State University. Sonya's research interests focus on web and software development, particularly in Web 2.0 supported social learning and virtual community, enterprise systems, agile development, usability, web analysis, and optimization. Sonya has published in multiple journals including IJEL, IJNVO, and JISE, and conference proceedings including ACM SIGMIS, AMCIS, GITMA, ICACC, ISOneWorld, ITNG, and SITE.



**Lorne Olfman** is a Professor in the School of Information Systems and Technology and Fletcher Jones Chair in Technology Management at Claremont Graduate University and former Dean of the School. Lorne's research interests include: how software can be learned and used in organizations, the impact of computer-based systems on knowledge management, and the design and adoption of systems used for group work. Lorne co-directs the Social Learning Software Lab (SL2). He has supervised 48 students to completion and has served on more than 50 other dissertation committees. He has co-authored more than 120 refereed articles including 42 in journals, 27 as book chapters, and 57 as conference papers.