

CS2830: Curriculum and Delivery Analysis 2011

Needs Assessment Summary

Prepared by: Justin Schuyler

Executive Summary

Recent advances in web development technologies have prompted the Computer Science Department at the University of Missouri to reconsider the content and role of Computer Science (CS) 2830: Introduction to the Internet, WWW, and Multimedia Systems. The existing curriculum is outdated, failing to prepare students for the current climate of web development. This Needs Assessment was conducted to reevaluate the content of CS2830's curriculum, establishing new learning objectives and determining what should and should not be taught. Furthermore, this Needs Assessment focused on CS2830's delivery method and investigated alternatives and improvements to increase the effectiveness of the lecture.

Data was gathered from a variety of sources including past curriculums, departmental policies, and observations of similar courses. In addition, two Computer Science instructors were interviewed via telephone to address the current state of the course, actuals, and the desired role of the course, optimal. Course content was researched and organized in a subject matter analysis according to the Knowledge and Cognitive Process Dimensions of Bloom's Revised Taxonomy. Finally, a force field analysis was used to decide the importance of online supplementary materials to CS2830.

Based on results of this Needs Assessment, it was recommended that CS2830 shifts its focus from site architecture to dynamic application development. This transition should address new technology in an effort to build better programmers. Additionally, this course should continue to be delivered as a lecture; however, it should increase the role of feedback, participation, and online supplementary materials.

Problem/Opportunity Statement

In the past year, web development technologies have undergone drastic changes. There's been a shift from XHTML to HTML5, from CSS2 to CSS3, and an increased emphasis on the use of JavaScript libraries (Appendix A). Additionally, web-based languages are gaining ground on desktop and mobile platforms, increasing CS2830's role in the Computer Science Department. In an effort to better prepare students for subsequent classes and teach the most contemporary and advanced web standards available, CS2830's curriculum must be reevaluated.

The first goal of this Needs Assessment is to determine what should and should not be covered by CS2830. Many of the recent advances in web technologies could be addressed in their own separate courses. This Needs Assessment will need to resolve the discrepancy between past expectations and new learning objectives. Content must be analyzed, organized, and either added or removed from the curriculum.

Furthermore, this Needs Assessment will need to evaluate CS2830's current delivery method. The existing format is a bi-weekly 75-minute lecture with supplementary notes and activities posted to a course website. This may or may not be the best way to administer the new curriculum. Alternative delivery methods must be considered, including independent study, online learning, and instructor supported lab work.

Failing to resolve the gap between past curriculums and new content would be detrimental to Computer Science and Information Technology students. Their education would lack significant updates in modern technology, which would disserve them in their future classes and potentially their future careers. In addition, failing to consider the delivery method may be deleterious to the effectiveness of the course altogether. The content may be sensible, but, if it is presented in an ineffective manner, then students will not be able to perform up to their potential.

The objective of this Needs Assessment is to determine what should be taught in CS2830 and how it should be taught.

Organizational Description

The University of Missouri is a not-for-profit institution composed of 10 colleges. The College of Engineering is devoted to the education and development of engineers as students, entrepreneurs, and future leaders in their respective fields. 113 faculty members from 9 disciplines teach approximately 2,900 undergraduate and 500 graduate students a year.

The Computer Science (CS) Department is led by 17 tenured faculty members teaching approximately 400 undergraduate and 110 graduate students a year. Their mission is to prepare students for Computer Science and Information Technology careers using state-of-the-art technology. The Information Technology degree program is one of 4 undergraduate programs offered by the CS Department. The CS Department is located in Engineering Building West on MU's campus. Courses are delivered as lectures or independent study. Online courses are not available to undergraduate students.

The CS Department is funded internally through the College of Engineering as well as externally through government and industry supported grants. The Information Technology program was created in 2005 via a donation from AT&T/SBC. Over the past year, increased interest and funding have led to the launch of a new website in addition to several changes in course curriculums and resources. IT courses are increasingly emphasizing the importance of interdisciplinary collaboration, establishing partnerships with other departments and schools.

Audience Analysis

The purpose of this Needs Assessment was to address the needs of the primary stakeholders in the content and delivery of CS2830, the students enrolled in CS2830. Each semester, approximately 50 undergraduate students enroll in this course. They are most often Computer Science or Information Technology majors; however, it is common to see 1 to 2 students from outside of the CS Department. Students are typically sophomores or juniors, given CS2830's prerequisite of CS2050, Algorithm Design and Programming II. They vary in age from 18 to 24 years old.

The average CS2830 student has at least 2 to 3 semesters of programming experience. They are responsible for understanding and being able to use fundamental programming concepts such as control structures and linear languages. Most students have minimal experience with web development.

Audience Analysis (cont.)

CS2830 students attend a 75-minute bi-weekly lecture on campus. An instructor introduces concepts and examples that are reinforced with online questions and mini-projects each week. A teaching assistant, typically a former CS2830 student, is available for student help. Students often collaborate, troubleshoot, and study with one another. They are responsible for learning the course material.

Primary and Secondary Data Sources

Primary Data Sources A Computer Science professor, Dr. Markita Price, and an Information Technology instructor, Matthew Dickinson, were interviewed as primary data sources for information regarding course content and delivery methods. Their previous experience teaching CS2830 allowed them to address fine details concerning the course's layout, progression, and overall learning objectives. Information Technology 3001: Collaborative Application Development using C# was also used as a primary data source for observations on course management and delivery.

Secondary Data Sources Secondary data sources included departmental information posted to the University of Missouri's Engineering website (<http://engineering.missouri.edu/cs/>) and past CS2830 curriculums (2009-2011) stored on Sakai (<https://sakai.missouri.edu>). The departmental information consisted of course requirements, graduation plans, and recommendations. The past curriculums included syllabi, examples, tests, and projects.

Data Gathering Techniques and Instruments

Extant Data Analysis The departmental information posted on MU's Engineering website and past CS2830 curriculums (2009-2011) stored on Sakai (Appendix B) were used in an extant data analysis. Information regarding the Computer Science Department was used to establish a context for CS2830 with respect to other CS and IT courses. This technique helped to clarify CS2830's role in the program. Past curriculums were analyzed to establish a precedent for historic expectations. They provided the exact topics, examples, and problems that past students were expected to learn.

Data Gathering Techniques and Instruments (cont.)

Telephone Interviews

Dr. Markita Price and Matthew Dickinson were interviewed via telephone (Appendix C) so as to gather information and insight concerning CS2830's past, current, and future role in the CS Department. They were selected based on their mastery of the subject matter as well as their current status as CS instructors. They teach courses that precede and follow CS2830, which affords a broad view of the educational scheme. They are also unique from other CS instructors in that they have taught CS2830 in previous semesters. This allowed them to address in-depth topics and examples covered by CS2830. The telephone interview was the most convenient and direct way to prompt for the information required by this Needs Assessment.

Subject Matter Analysis

Dr. Price, Matthew Dickinson, and the past curriculums explored by the extant data analysis were all used as resources for a subject matter analysis (Appendix D). This technique was used to fully understand the content of CS2830 and the implications of adding or removing topics. It established the necessary background for making informed decisions on these updates.

Observation

Observations of InfoTech3001 were taken for comparison (Appendix E). Like CS2830, InfoTech3001 has experienced recent updates in its course curriculum. The goal of these observations was to explore InfoTech3001's reaction to changes and use that information to generate optimal for CS2830's transition. Observation was also used to directly note the effectiveness of InfoTech3001's delivery. It was chosen as a technique because of the detail it provides beyond that of an interview.

Force Field Analysis

A force field analysis was used to help evaluate the advantages and disadvantages of expanding the role of online supplementary materials in CS2830 (Appendix F). The weighted results were used to suggest changes to CS2830's delivery method. This technique was chosen for its purpose as a well-organized decision-making tool. The interview and observation results were the primary data sources for this analysis.

Data Gathering Process

*Extant
Data
Analysis*

Extant data regarding departmental policies and course context were researched by browsing MU's Computer Science website. The information is organized based on 4 degree programs. Each program section contains course requirements, descriptions, graduation plans, and recommendations for enrollment. This data was read and analyzed to define CS2830's role in the CS/IT program. To review data regarding CS2830's course curriculum, it was necessary to gain access to the course website, Sakai. Gaining access was not difficult, but it took time and required getting approval from the System Administrator. Once accessible, the curriculums for multiple semesters were reviewed and analyzed for content, order, and learning objectives.

*Telephone
Interviews*

A Computer Science professor, Dr. Markita Price, and an Information Technology instructor, Matthew Dickinson, were interviewed via telephone. The purpose of these interviews was to determine their views on the current course content and instruction modality for CS2830. The interviews were conducted at a pre-agreed upon time in the afternoon. The Interview Protocol consisted of an introduction, a series of organized questions, and a conclusion. After the interviews, answers were compared and contrasted.

*Subject
Matter
Analysis*

The interviews and course curriculums were used as resources for the subject matter analysis. The topics and skills covered by CS2830 were organized and classified according to Bloom's Revised Taxonomy (Anderson & Kratwohl, 2001). Each curriculum item, both existing and potential, was evaluated based on the Knowledge and Cognitive Process Dimensions. This provided an outline for the focus, scope, and direction of the class.

Observation

Two InfoTech3001 classes were observed in a week. An observation procedure was developed to organize and direct the observer's attention. Notes were recorded on paper and included details about course characteristics (size, location, mood), content (paradigms, difficulty, scope), and delivery (supplemental materials, instruction, interaction). These notes were organized at the end of class.

*Force
Field
Analysis*

The force field analysis was conducted towards the end of this Needs Assessment. It took into account answers from the telephone interviews and observations of InfoTech3001 to generate a list of pros and cons facing the increase of supplementary online materials. Once a list was compiled, each advantage and disadvantage was assigned a weighted value to represent its importance in the context of CS2830. The weighted values were added and the totals were compared to decide whether or not to expand the role of online materials.

Data Analysis Process

Extant Data Analysis

The extant data was primarily non-numeric. It consisted of departmental policies, graduation plans, and enrollment recommendations. This information was read and compiled to produce an accurate background context for CS2830 and its role in the CS Department. The extant data regarding course curriculums, tests, examples, and projects was analyzed by comparing the progression of topics covered from semester to semester. Advances in technology were reflected in updates to more recent curriculums. These updates were analyzed based on their scope, content, and overall impact on the course.

Telephone Interviews

The telephone interview results were analyzed after the interviews were conducted. Notes recorded on the Interview Protocol were expanded and summarized into an organized report. The answers from each interviewee were compared and contrasted to group similar ideas and explore conflicting ones. One of the difficulties with conducting a post-interview analysis was that it eliminated the possibility of clarification. If the analysis yielded more question than answers, it was not possible to go back and re-ask those questions. They would have to be organized and featured in a follow-up interview.

Subject Matter Analysis

The subject matter analysis yielded several topics and skills that should be covered in CS2830. These items, both existing and potential, were organized and classified according to Bloom's Revised Taxonomy (Anderson & Kratwohl, 2001). This required analyzing each curriculum item and determining its relationship with the Knowledge and Cognitive Process Dimensions. The Knowledge Dimension features 4 groups (Factual, Conceptual, Procedural, and Meta-Cognitive) whereas the Cognitive Process Dimension features 6 categories (Remember, Understand, Apply, Analyze, Evaluate, and Create). The only difficulty with using Bloom's Revised Taxonomy was that a couple of the topics and skills could have been applied to more than one of the dimensions. Part of the analysis required determining the best dimension to describe the given topic or skill.

Observation

The observation data was analyzed after each class by expanding truncated notes and grouping otherwise chronological events. After the second class, all of the observations were compiled and analyzed for similarities. They were organized and explained in a summary report.

Force Field Analysis

The force field analysis required a detailed investigation into the pros and cons of online supplementary materials. This involved compiling and analyzing the results from the interviews as well as the observations of InfoTech3001. Once an exhaustive list was created, each advantage and disadvantage had to be evaluated individually and assigned a weight value. This value determined the importance of that factor and added to a total, which ultimately favored an increase in online materials.

Data Analysis Process (cont.)

The difficult part associated with a force field analysis is the subjective nature of assigning weight values. Many factors could be weighted higher or lower given a different context than CS2830.

Data Table

Technique	Instrument	Source	Summary
Extant Data Analysis	Computer Science Department Policies and Information at MU	Departmental website: http://engineering.missouri.edu/cs/	CS2830 is a mid-level undergraduate course for CS and IT sophomores and juniors. It serves as a transition between the programming fundamentals of CS2050 Algorithms II and the advanced topics of CS3380 Database I and CS4320 Software Engineering.
Extant Data Analysis	Sakai CS2830 Curriculums	Online CS2830 curriculums from 2009-2011: https://sakai.missouri.edu (Includes syllabi, tests, examples, projects)	The focus of CS2830 has, in this semester (fall 2011), shifted from website structure, templates, and procedural server communication to dynamic application development. Extensible HyperText Markup Language, PHP, and Cascading Style Sheets 2 were the core tools used in the course. There is an increasing emphasis on online tools and usability interaction.
Telephone Interview	Interview Protocol	Dr. Markita Price	CS2830 should consider presenting examples that could be worked on simultaneously by students and the instructor. Feedback should be improved. The content of the course should better support subsequent courses like CS4320 Software Engineering. Finally, usability should be reinforced throughout the class.

Data Table (cont.)

Technique	Instrument	Source	Summary
Telephone Interview	Interview Protocol	Matthew Dickinson	CS2830 should remain a lecture, but increase its focus on online supplementary materials. Ideally, students would be involved in the creation of tutorial resources. Students should learn how to solve problems using documentation rather than memorization. The core technologies covered should be updated to HTML5, CSS3, and JavaScript.
Subject Matter Analysis	Bloom's Revised Taxonomy	Telephone interview and course curriculum results	CS2830 is responsible for teaching students the fundamentals of web design, engineering, and conceptual programming models; although, its primary goal is to produce better programmers, not just web developers. The majority of the content covered in CS2830 falls under the 'Understand' and 'Apply' categories of the Cognitive Process Dimension. They also largely represent the 'Conceptual' and 'Procedural' groups of the Knowledge Dimension. Newer technologies including HTML5, JavaScript, and CSS3 should be integrated into the curriculum.
Observations	InfoTech3001 Observation Protocol	The instructor and undergraduate students of InfoTech3001	InfoTech3001 has adapted to changes in its curriculum by stressing the importance of documentation over memorization and providing additional online supplementary materials. The course is still delivered as a lecture, but it has successfully integrated an open dialogue between the instructor and students. In addition, course examples are presented step-by-step as problem-solving opportunities.
Force Field Analysis	Force Field Analysis Protocol	Telephone interview and observation results	CS2830's online supplementary materials should be expanded.

General Data Summary and Interpretation

The data collected by this Needs Assessment suggests that CS2830 update its curriculum to address the emergence of new technologies, improve its role as part of the Computer Science Department, and stress the importance of problem-solving via documentation. The extant data analysis defined CS2830's role as an extension of programming fundamentals and a transition to more advanced software engineering. Past curriculums have been updated, but not drastically changed until now. The interviews revealed a need for modern web developing tools, such as HTML5, CSS3, and JavaScript. In addition, the interviewees agreed that CS2830 should be delivered in a lecture format, but that it should involve greater interaction between instructors and students. Feedback should be improved and online supplementary tools should be expanded to help students meet the learning objectives. The force field analysis and observations reaffirmed the necessity for increased online materials. Finally, the subject matter analysis provided a technical background for the purpose of CS2830. Students should learn the basics of web design, engineering, and conceptual programming models; however, the primary goal should be to produce better programmers, not necessarily web developers.

Knowledge/Skill Needs-Details

The skillset taught by CS2830 must be updated to reflect advances in modern web development technologies. Students need to know less about site structure and more about the creation, engineering, and troubleshooting of dynamic web applications.

Specifically, they need a curriculum that teaches them the following:

- The fundamental structure and syntax of modern web developing tools such as HTML5, CSS3, and JavaScript.
- The conceptual models of client/server communication, Model-View-Controller, and Object Orientation
- The ability to design, create, and troubleshoot dynamic web applications
- The problem solving skills required to answer foreign questions through documentation

When confronted with a web development problem, students need to understand what resources are available. They need to be able to efficiently discern between multiple resources to find relevant and accurate information. Once that information is discovered, they need to apply their findings towards solving the problem. Often times, students will struggle at this point because they are not accustomed to analyzing code. They are more likely to copy code exactly than they are to read the code, understand it, and build their own implementation. A sample learning objective would be to present students with a small array of partial solutions and then require them to combine those solutions to create a greater whole. Students need to understand code at a conceptual level, not just a procedural level.

Recommendations

CS2830's curriculum should be updated to address recent advances in modern web development technology and better fulfill its role as part of the Computer Science degree program. To effectively help students achieve new learning objectives, the delivery method should be reorganized to promote feedback and provide supplementary examples, notes, and exercises. CS2830's new curriculum and delivery method should support the following skills and objectives:

Build Better Programmers

While teaching the fundamentals of web languages and concepts represents a large portion of CS2830, the goal should always be to build better programmers, not necessarily better web developers. Paradigms should be extended to programming in general and used to encourage best practices. Problem-solving should be the crux of every assignment and test question. In addition, students should be encouraged to apply principles of usability and become aware of interaction design.

Address New Technology

New web development technologies (Appendix A&B) must be addressed to ensure that the course is teaching state-of-the-art practices and capabilities. Old concepts of site architecture, templates, and procedural server communication should be replaced or compromised for sections on dynamic communication and application development.

Increased Feedback and Supplementary Materials

CS2830's delivery should focus on feedback and open dialogue between students and the instructor. Examples should be designed to encourage students to participate and ask questions. Supplementary material should be offered online. These resources should include additional examples, tutorials, and links. Ideally, students should take part in the creation, suggestion, and maintenance of these resources. Providing an opportunity for participation allows for a sense of ownership. Altogether, these resources should increase the effectiveness of the lecture and provide support for students needing additional help.

Appendix A – Advances in CS2830 Curriculum Topics

2010 Technology	2011 Technology	Notes
XHTML	HTML5	HTML5 builds upon XHTML's syntax and structure, but expands functionality and cross-platform compatibility. Flash is steadily being replaced by HTML5.
CSS2	CSS3	CSS3 extends traditional cascading stylesheet methods, affording web designers greater freedom of expression.
JavaScript	Asynchronous JavaScript and XML	JavaScript is used to add control/behavior to a webpage. Asynchronous JavaScript and XML (AJAX) allows for dynamic communication with a server. It was not covered by CS2830 until the fall 2011 semester.
NA	jQuery and jQuery User Interface	jQuery is a popular JavaScript library that was also not taught in CS2830 until the fall 2011 semester. jQuery UI provides an opportunity to discuss user interface and the importance of usability in interaction design.
NA	Mobile Application Development	Adobe, Microsoft, and Google have started to push the role of HTML5, CSS3, and JavaScript in the development of mobile applications.

Appendix B – Computer Science 2830 Curriculum Fall 2011

Class #	Date	Topic
1	23-Aug	Intro to Course, Basic HTML and CSS
2	25-Aug	CSS Box Model and Floats
3	30-Aug	CSS3 Multi-column layout
4	1-Sep	CSS Selectors and CSS location and Priority
5	6-Sep	Tables
6	8-Sep	Transitions and Animations
7	13-Sep	Forms
8	15-Sep	Start Javascript
9	20-Sep	DOM and Access of form data
10	22-Sep	Jquery Intro
11	27-Sep	Building HTML5, CSS3, and JS Apps
12	29-Sep	Same as 27th
13	4-Oct	Web Server Intro
14	6-Oct	Intro to PHP
15	11-Oct	PHP arrays, form data (post and get)
16	13-Oct	Client/Server Data Communication
17	18-Oct	Object Orientation in PHP
18	20-Oct	MIME-Types and Retrieving different kinds of data
19	25-Oct	File Upload
20	27-Oct	GD (graphics) and php graphics library
21	1-Nov	Ajax
22	3-Nov	Web Services
23	8-Nov	Jquery UI: Building interfaces with Jquery
24	10-Nov	Login/Authentication and security
25	15-Nov	Database I
26	17-Nov	Database II
27	29-Nov	PHP Templates
28	1-Dec	Rich Text Editors
29	6-Dec	Jquery Plugins and Javascript Closures
30	8-Dec	The Future Web Technologies

Appendix C – Telephone Interviews

Interview Protocol

Introduction

About Me: My name is Justin Schuyler and I am a graduate student at the University of Missouri’s School of Information Science and Learning Technologies.

About Project: I am conducting this interview as part of a needs assessment to address changes to computer science 2830, introduction to web development.

Interview Goal: The purpose of this interview is to help determine what should be included in CS2830’s course content and how this content should be delivered to students. I will ask you to share your expertise and experiences teaching as well as your thoughts on the changes occurring in CS2830. The interview should last approximately 15 to 20 minutes. Do you have any questions?

Interview Questions

Tools	What methods of instruction have you used in your teaching career? Lectures, independent study, e-learning? In your experience, what teaching methods have been most effective? If you had to rank each method, how would they compare to one another with respect to effectiveness?
Environment	What is the best environment for students learning computer science? Can this be accomplished via a lecture or is individual attention required? What would be the best environment for learning web development?
Capacity	Does this argument apply regardless of class size? Given the current class size of 55 students in CS2830, is a lecture the most effective method of teaching?
Vision	How should CS2830 be taught? What should be the primary focus of CS2830? What technologies should be explored in CS2830? To what extent?

Expectations	<p>What should students be capable of after taking CS2830?</p> <p>For the following categories, what should and should not be covered in CS2830?</p> <ul style="list-style-type: none"> • Basic web structure • New web standards • General web design • Scripting and page behavior • Database management
Feedback	<p>As a teacher in the program, are you confident in students' web development abilities? What do they do well and what do they need help with? What is a good way to measure expectations?</p>
Processes	<p>How do you handle updates in technology with regard to determining what should be taught?</p> <p>How frequently should CS2830's lesson plan be updated? Rarely? On occasion? Frequently?</p>
Vision Continued	<p>If one thing could be done to improve CS2830, what would it be?</p>

Conclusion:

- To summarize, we discussed _____ and you said _____
- Do you have any questions or additional comments?
- Thank you for your time and effort participating in this interview.

Data Summary

Both interviewees agreed that the current lecture's format lacks some degree of effectiveness. As noted by the first interviewee, the lecture is a cost-effective teaching method commonly used by the department. It is an adequate tool in general, but much more successful when paired with or accented by additional teaching modalities. One example given was the idea of student-instructor interaction. When teaching smaller classes at another university, the first interviewee mentioned the convenience of working on the same code with students simultaneously. Even though the CS2830 lecture is not held in a computer lab, the first interviewee argued that enough students have laptops that the current lecture room could become a computer lab upon request. The second interviewee echoed a similar point, stating that the vision for CS2830 is to keep a lecture for overall guidance and motivation purposes, but accent it with interactive, e-learning tools. Online resources would be especially helpful as technology continues to change and students need a place to gain meaningful practice.

Both interviewees mentioned the following concepts as topics that should be covered in CS2830:

- Fundamental Web Development Concepts (JavaScript, HTML5, CSS, PHP, MySql)
- Web design and behavior (via the Model/View/Controller paradigm)
- Human Computer Interaction
- Usability

It was stressed in each interview that the content for CS2830 should approach web design from a current, state-of-the-art perspective; however, its core should revolve around the application of fundamental web technology and the concept of building new skills. The purpose of the class is not to memorize a set of commands or control structures, but rather, it is to explore documentation and prepare students to apply their knowledge and procure new knowledge as web development continues to change.

Appendix D – Subject Matter Analysis

Bloom’s Revised Taxonomy

The Knowledge Dimension	The Cognitive Process Dimension					
	Remember	Understand	Apply	Analyze	Evaluate	Create
Factual		2. Web Terminology 3. Syntax				
Conceptual		4. Client/Server Model 5. Web Services 6. Object Orientation 7. Security	9. Model-View-Controller 10. Site Structure	14. Document Object Model		
Procedural	1. Control Structures	8. CSS/jQuery Selectors	11. Language Structure 12. Server Communication 13. Database Communication		16. Evaluate Solutions	17. Create Solutions 18. Create Final Project
Meta-Cognitive				15. Analysis of Documentation		

Remember

1. Control Structures [Procedural]: students are expected to recognize and recall basic techniques for accomplishing programming tasks. Examples include methods for loops, sub-procedures, and if-then statements.

Understand

2. Web Terminology [Factual]: students will be able to explain web-related problems, designs, and solutions using basic web-terminology.
3. Syntax [Factual]: students will learn how various web-languages are structured (including HTML, CSS, JavaScript, PHP, MySQL, and more). They will compare basic terminology and summarize key characteristics.
4. Client/Server Model [Conceptual]: students will be capable of generalizing the relationship between clients and servers. They will categorize and explain web interactions based on the Client/Server Model. They will understand what types of communication are available.
5. Web Services [Conceptual]: students will exemplify their understanding of the Client/Server Model by using it to abstract the purposes of web services. Web services will be introduced conceptually, rather than explored at the procedural level. Students will be able to infer relationships between services.

6. Object Orientation [Conceptual]: students will understand the concept of object orientation, why it is used, and its structure. They will be able to compare and contrast procedural code from object-oriented programming.
7. Security [Conceptual]: there is not enough time to address security procedurally, so it is highlighted periodically as a concept. Students are expected to understand the principles of security, why it is important, and what to look for. They should be able to predict potential security vulnerabilities.
8. Selectors [Procedural]: students will understand how selectors work and how they can be used in design. They will be able to compare different types of selectors and explain where, when, and how each type should be used.

Apply

9. Model-View-Controller [Conceptual]: students are taught the Model-View-Controller Paradigm as a method for organizing their code. They are expected to apply MVC in all of their assignments.
10. Site Structure [Conceptual]: students learn how websites should be organized. The concepts of site structure, reusable code, and templates are covered. Students are expected to implement these concepts in their projects.
11. Language Structure [Procedural]: students learn the techniques, methods, and algorithms behind a variety of web languages. They apply their skills through problem-solving exercises.
12. Server Communication [Procedural]: students are taught the steps to and requirements for communication with a server. They apply this knowledge by implementing several different communication techniques (HTTP Requests, AJAX, POST/GET, and more).
13. Database Communication [Procedural]: students learn how to connect to a database and make queries. They implement these procedures in their final project.

Analyze

14. Document Object Model [Conceptual]: students are capable of organizing a webpage with respect to the Document Object Model. They can deconstruct a page into interrelated pieces that all serve a purpose.
15. Analysis of Documentation [Meta-Cognitive]: one of the most critical lessons taught by CS 2830 is the importance of documentation. Many students are unfamiliar with or have never used documentation as a problem-solving tool. Even fewer have experience navigating Application Programming Interfaces. CS 2830 strives to change the way that students approach programming. Part of that involves increasing awareness of how

students solve problems and where they go for help. Another part involves planning and breaking down documentation.

Evaluate

16. Evaluate Solutions [Procedural]: students observe different solutions for the same problem and determine which solution is best for a given situation.

Create

17. Build Solutions [Procedural]: students plan and create their own methods, algorithms, and solutions for in-class problems. They put everything together to produce a functional project.
18. Create Final Project [Procedural]: students combine their cumulative knowledge towards a final project. They apply various skills, methods, and techniques to plan, design, and produce a functional website.

Appendix E – InfoTech3001 Observation Protocol Collaborative App Development using C#

Date: 10/7/2011

Time: 1pm – 1:50pm

Location: 237 Engineering Building West

Number of students: 8

Guiding Prompt: How does InfoTech 3001 adapt to updates in the course curriculum? Look for tools, feedback, knowledge assessment, incentives, and overall structure.

Observation Category	Details	Potential Optimals	Additional Notes
Processes + Tools			
Knowledge + Skills			
Environment			
Feedback			
Motivation			
Other			

Results

Guiding Prompt: How does InfoTech 3001 adapt to updates in the course curriculum? Look for tools, feedback, knowledge assessment, incentives, and overall structure.

Observation Category	Details	Potential Optimals	Additional Notes
Processes + Tools	<p>At the beginning of class, the instructor shifted a due date based on feedback from the students.</p> <p>Due to students' unfamiliarity with the new development environment, time was spent to go through step-by-step examples of how to accomplish basic functions.</p> <p>Students aren't taught to memorize concepts, but rather, they are shown where to go to find the answers.</p> <p>Online resources were used to accent the content in the lesson.</p>	<p>Flexibility may be critical with the introduction of new/foreign material.</p> <p>To address the introduction of new coding environments, it can be useful to invest time into step-by-step examples that show students the basics.</p> <p>Adaptation and perseverance are more important than rote knowledge. Students should be encouraged to look up documentation rather than memorize it.</p> <p>Supplementary material can accent student learning.</p>	<p>The instructor was flexible, but reiterated that the extension was not for the sake of "slacking off", but to "really explore" the lesson.</p> <p>Example: the instructor reviewed an example using a student's code to demonstrate how to debug an application.</p> <p>Example: several times during the lecture, the instructor walked students through the process of looking information up. He prompted students to do the same, rather than reciting from memory.</p> <p>The online learning management system used is Moodle.</p>
Knowledge + Skills	<p>Functional code snippets were used to illustrate points, rather than generating code from scratch. The instructor probed students' understanding of the code via a "cat and mouse" line of questioning.</p>	<p>Prepared code may be the best way to show students how a general concept works without wasting time on small details.</p>	<p>Using prepared code allows the instructor to gauge the students' knowledge/skill level. By asking questions, they can determine what concepts are clear and what should be reviewed.</p>

Environment	Each student has a PC. The PCs are all configured the same way, so the students can follow along using the same hardware and software as the instructor.	As a programming class, it is important for students to get hands-on practice with new code. Additionally, continuity can be critical. It is easier for students to grasp the material when they don't have to worry about differences in hardware/software.	Reaffirming the point of continuity, a comment was made by the instructor on the compatibility of a particular application in that day's lesson. The issue was relevant to each student due to the platforms they were all using.
Feedback	Class is conducted partially as an open dialogue between instructor and students. Students are encouraged to ask questions and make comments. The instructor took the time to address each comment with thorough feedback.	Feedback and dialogue is important to increase interaction between the instructor and his/her students. This allows the instructor to gauge knowledge/skills, delve into new topics, or explore explanations.	Example: One student mentioned that the behavior of an aquarium application would be better if you could add more varied movement to the characters. The instructor praised the idea and showed 3 ways to accomplish it.
Incentives + Motivation	The last 15 minutes of the class was spent helping individual students debug/fix their code. Students without questions were allowed to leave.	The incentive of getting one-on-one help for a class of 8 students is appropriate; however, this is not applicable to a larger course like CS2830. Nevertheless, it can be motivating for students when time is allocated for addressing questions, concerns, and/or examples that they are proud to show.	Respecting students' time and priorities motivated them to stay and get help or help others around them.
Other	Humor was used repeatedly to lighten the mood of the class and even inspire creativity.	Humor is difficult to prescribe as an optimal; although, the point to stress here might be fluid transitions . In this case, humor was used to help encourage creativity and move on to new topics.	Example: students were encouraged to discuss what other random objects they might include in the aquarium application.

Appendix F – Force Field Analysis

What are the advantages and disadvantages of expanding online supplementary materials?	
Advantages	Disadvantages
Online materials provide students with the opportunity to review course material. This can reinforce or supplement the primary delivery method (whether it is a lecture, independent study, etc.) 3	Online supplementary materials risk diminishing students' reliance on the primary delivery method. If the notes and examples are posted online, for instance, students may be inclined to shift their attention away from the lecture. 1
Multiple mediums are available for instruction. Materials may come in the form of text, audio, video, charts, graphics, news articles, PowerPoint presentations, and more. 3	Online materials are expensive to prepare and maintain. Multimedia resources like audio and video require extra time and effort to develop. 2
Online resources are extensible. A knowledgebase of course material can be used, re-used, and added to over several semesters. 2	These costs may not be worthwhile if the resources created have to be updated or redone on a consistent basis. 2
An online environment provides opportunities for collaboration and open dialogue. Whereas student participation in large lectures is limited, every student has the option of contributing to an online forum. 2	Student assessment and feedback can be limited. In large classes especially, the instructor may not be able to respond to every comment, discussion, or question. Feedback costs additional time and energy that might be better spent elsewhere. 2
Online materials are highly accessible. Instructors have the flexibility of providing students with tasks, instructions, and lessons in a centralized and organized manner. Students have the ability of reviewing these resources in any place and at any time. 2	One of the biggest threats to online learning is motivation. The success or failure of additional online resources depends on the impetus of student interaction. Materials are accessible, but that doesn't ensure that they will be accessed. 3
Using online materials is appropriate for a class that focuses on web development. Most of the resources, problems, and examples used by the course are web based/accessible. 1	Expanding the role of online materials assumes that students are comfortable using and willing to participate in those materials. 1
Online forums can break down gender, racial, cultural, and personality barriers. They can encourage participation that might not otherwise take place. 2	Online forums can create new social barriers. Students may not feel comfortable using online materials as a medium for communication. Intentions, tone, and voice can be lost in electronic mediums. 2
Total: 15	Total: 13