Grading Technology Allows Teachers to Infuse Technology in the Economics Classroom

Ian J. Shepherd
Abilene Christian University

Brent Reeves
Abilene Christian University

Darryl Jinkerson
Abilene Christian University

Education courses now involve homework assignments that require technology skill as well as domain knowledge. Yet there is little pedagogical and technological support for teaching "What" (statistical mean) while simultaneously teaching "How" (use the =average (Range) function in Excel). We describe a conceptual approach and a methodology that helps teachers leverage their domain knowledge and helps students learn both a new topic and a new information technology skill. While teachers will allocate more time towards preparing homework, far less time is spent overall in administering and grading assignments. This approach scales to any class size, thus removing grading burdens imposed by large class sizes. The huge burden of grading lessons is removed, leaving that time free to improve the teaching.

INTRODUCTION

Higher education courses now involve homework assignments that require skill at a technology as well as an understanding of a domain concept. Yet there is little pedagogical and technological support for teaching "What" (Average) while simultaneously teaching "How" (use the Average function in Excel). This paper describes a conceptual approach and a system implementation that helps teachers leverage their domain knowledge and helps students both learn a new topic and new information technology skill. While teachers might allocate more time towards preparing homework, far less time is spent overall in administering and grading assignments. This approach scales to any class size, thus removing grading burdens imposed by large class sizes.

Large class sizes have made it more difficult for teachers to provide individual feedback and attention to each student [Chamilliard 2002, Meiselwitz 2002]. It is not unusual at the university level to have sections of classes with hundreds of students [Kay 1998]. These large classes, while financially lucrative for the schools, cause concern for teachers because they can no longer provide one-on-one feedback to students.

Large classes have led to a change in teaching philosophy from the Socratic Method where the professor operates in a mentoring type environment, to straight lecture, where the topic is presented at a pace and with a style that makes little room for individual learning styles of students. One-on-one interaction is limited and individual feedback is rare. Piaget maintained that individuals learn through interaction with the real world and that social interaction develops knowledge [Piaget 1969].

The Socratic process provided an individualistic teaching methodology. Students were prompted with questions to explore and develop their own understanding of the topics at hand. Individual questioning by the teacher guided the learners to new levels of understanding. The Socratic method of teaching was seen as a powerful tool in developing critical thinking through self-discovery. The teacher used guided questions to develop the thinking skills of the student. Each question was specifically designed for that individual to assist in the development of their learning skills.

Instructional technologies have been limited in their ability to gauge an individual's progress and offer the teacher the ability to iteratively guide the student towards new propositions using the manipulation of information. Traditional methods of instruction would require significant teacher time commitments and exceptional time management techniques to provide unique one-on-one feedback and guidance through a series of questions. The authors' prototype developments, however, will provide tools that are capable of providing one-on-one instructional feedback to unlimited number of students.

Marketplace pressures have also resulted in recent curriculum changes at all school levels. More emphasis is being placed on the integration of information technology in all courses [Horgan 1998]. It is typical for visiting committees and school boards to recommend spreadsheet and database software are integrated throughout all curriculums. So in addition to the usual domain knowledge (accounting, finance, management, BCIS, statistics), the student must also learn database, spreadsheet, presentation, data-mining software etc.

Common approaches to integrating technology in the curriculum include requiring introductory courses in, for example, Microsoft ™ Excel. Although one must begin somewhere and the learning curve of most current information technology tools is steep, research shows that the ideal learning environment is in the context of real problems [Suchman 1987]. If the best learning takes place in context of a real problem, an ideal Macroeconomic assignment would introduce new spreadsheet concepts in the context of an economics issue, for example elasticity. The student would improve their skill set (spreadsheet knowledge) as well as their interpretation skill (elasticity).

THE PRIMARY GOAL - DOMAIN KNOWLEDGE

The primary goal of an assignment is to teach domain knowledge. The student is challenged to demonstrate their new knowledge in the context of some problem. Two things interfere with this. First, the increased pressure to infuse technology interferes with domain learning. The learning curve of desktop software applications is steep so much time must be allocated to learning the technology itself. This time and energy can detract from learning about the domain. Second, learners make two kinds of errors: syntactic and semantic [Histova 2003].

Syntactic Errors

A syntactic error is frequently referred to as a "typo" or typographical error. An example of a basic syntactic error is when a spreadsheet user forgets to type "=" before entering a formula. The spreadsheet software doesn't recognize the text "A1+B1" as something to be calculated, but as text to be displayed. A more insidious syntactic error is when the formula is "correct" only in the sense of being accepted by the spreadsheet software. In our context, an example is when a student intends to write =A1 + B2, but instead, enters =A1 – B2. The plus and minus keys are side-by-side on many keyboards and is easy to mistype these keys.

The challenge that syntactic errors introduce in technology intensive courses is that they confuse the semantic issues. It is difficult to understand elasticity when the formulas are not correct. A small typographical error can cause much confusion.

Semantic Errors

Semantic errors are true misunderstandings. An example is when a student does not understand how to calculate a slope. The student enters a formula which produces erroneous results. Spreadsheet software cannot know that the formula was intended to calculate "slope," so there is no way to catch this error other than to recognize an incongruity between expected and actual values. Misunderstandings at the semantic level can cause a student to spend much time adjusting formulas that are technically correct, but not appropriate.

Information Technology Skills

The marketplace has increased demand for business graduates with skills in desktop software applications. Most often this is the Microsoft ™ Office suite, but there is also more interest in SAS, SPSS and SAP. The intent of this article is to focus on spreadsheet assignments using Microsoft ™ Excel.

The System

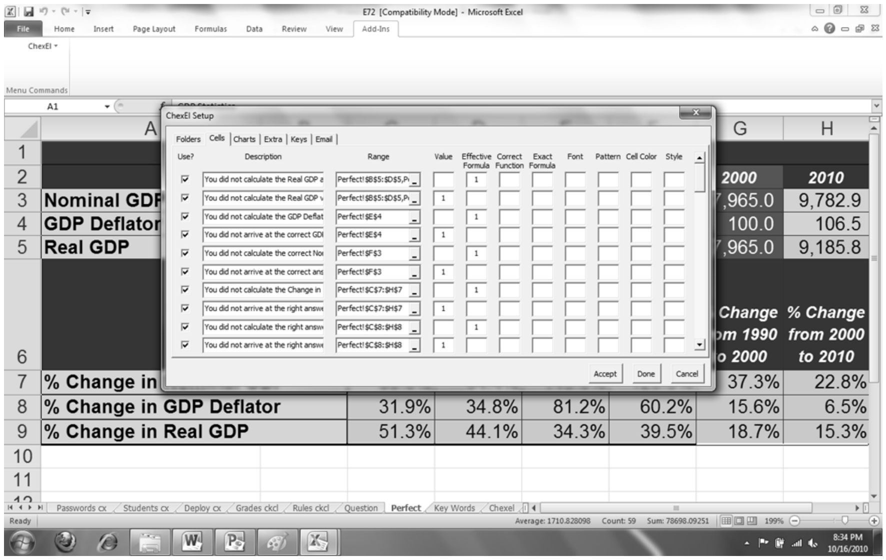
In design and development for twelve semesters by the authors, the original desktop system managed the distribution, grading, and feedback of spreadsheet homework assignments. The prototype system was designed around some simple steps:

FIGURE 1
STEP 1 – THE PERFECT ANSWER

GDP Statistics							
	1950	1960	1970	1980	1990	2000	2010
Nominal GDP	275.0	549.0	1,067.0	2,596.0	5,803.2	7,965.0	9,782.9
GDP Deflator	16.8	22.1	29.8	54.0	86.5	100.0	106.5
Real GDP	1,641.8	2,484.2	3,580.5	4,808.6	6,708.9	7,965.0	9,185.8
		% Change from 1950 to 1960	% Change from 1960 to 1970	% Change from 1970 to 1980	% Change from 1980 to 1990	% Change from 1990 to 2000	% Change from 2000 to 2010
% Change in Nominal GDP		99.6%	94.4%	143.3%	123.5%	37.3%	22.8%
% Change in GDP Deflator		31.9%	34.8%	81.2%	60.2%	15.6%	6.5%
% Change in Real GDP		51.3%	44.1%	34.3%	39.5%	18.7%	15.3%

The professor prepares a template containing the perfect answer and decides what is important in this assignment. By spending a little more time on the assignment, we can test different levels of learning, both semantic and interpretive kinds of understanding. This preparation takes more time than before, but our experience shows that, for example, with a class of 200 students, the grading time is reduced by over 90% [Shepherd 2005].

FIGURE 2
STEP 2 – CREATE GRADING RULE CRITERIA



The teacher creates grading rules for the perfect answer. The desktop tool has checkboxes the teacher uses to indicate which aspects of the assignment should count. The teacher can also assign grade weights to each criteria being checked. Our context being spreadsheets, the desktop set-up options include formula, value, and cell attributes such as font, style, or colors. The software can focus on syntactic issues: is the formula correct? Is the answer correct? Is the data shown correctly? The system can also focus on semantic issues: what data meets a certain criteria? What does this chart mean? How might this be interpreted?

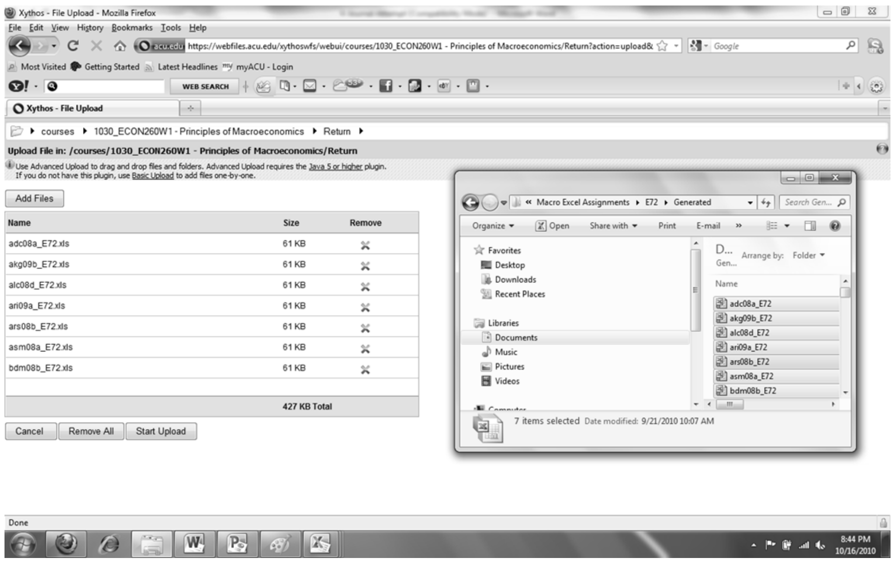
Additional time during rule development ensures clear grading criteria are maintained. It is during this process that the teacher can decide on the level of feedback to each student. Assignment intent and teaching philosophy are handled by allowing the teacher to provide simple feedback: “This is wrong – fix it” to “You did not calculate the average correctly. To do this you need to….”

FIGURE 3
STEP 3 – CREATE A STUDENT TEMPLATE FOR DISTRIBUTION

GDP Statistics							
	1950	1960	1970	1980	1990	2000	2010
Nominal GDP	275.0	549.0	1,067.0	2,596.0		7,965.0	9,782.9
GDP Deflator	16.8	22.1	29.8		86.5	100.0	106.5
Real GDP				4,808.6	6,708.9	7,965.0	
		% Change from 1950 to 1960	% Change from 1960 to 1970	% Change from 1970 to 1980	% Change from 1980 to 1990	% Change from 1990 to 2000	% Change from 2000 to 2010
% Change in Nominal GDP							
% Change in GDP Deflator							
% Change in Real GDP							

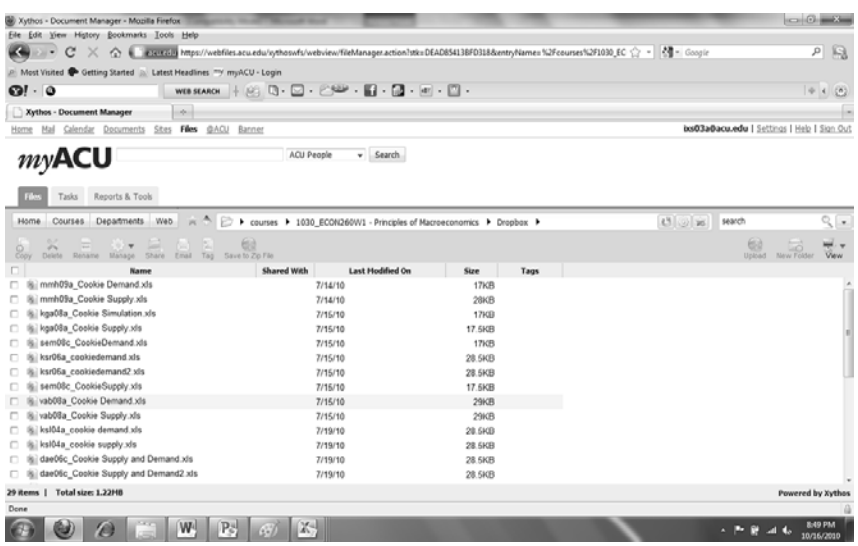
Having finalized assignment creation, the teacher creates a blank template by removing from the perfect answer those items to be completed by the student. Instructions are clearly given as to what the student must complete to receive full assignment credit.

FIGURE 4
STEP 4 – DISTRIBUTE THE ASSIGNMENT TEMPLATE FOR COMPLETION



In the prototype system the blank template assignment was distributed to the student via common directory, email attachment, or drop/return box systems. The new web based version of the product, removes this step by allowing assignments to be downloaded from the web. The new web based distribution system removes all local architecture problems for teachers. Common barriers to mass distribution of the prototype were: lack of an email system to send these files out or, lack of file distribution system, difficulty processing files by email attachment. Now, all that is required is access to the web- not infrastructure is necessary for the school.

FIGURE 5
STEP 5 – THE STUDENT COMPLETES THE ASSIGNMENT



After downloading the assignment, the student uses the usual spreadsheet software, in our case, Microsoft Excel, and then submits the file to the school dropbox system.

FIGURE 6
STEP 6 – GRADE THE ASSIGNMENT

ChexEl (cc828)		10/16/2010				14		You did not arrive at the correct answer for Nominal GDP	You did not arrive at the correct GDP Deflator Value
			Assignment Value	10				Value (1 pt.)	Value (1 pt.)
			Possible Points	12			Cells:	1	1
E72 Excel Activity							Pts/Cell:	1.00	1.00
							Correct:	0.50	0.50
Id	*Id2*	*Last*	*First*	*Email*	*Grade*	*%*	*Points*	*PERFECT F3*	*PERFECT E4*
00042d9323	E72 Assignment	Snapper	Ashley	ac08a@acu.edu	5.56	56%	6.67	0	0
000a432522	E72 Assignment	Goldsmith	Bobby	ag09b@acu.edu	10.00	100%	12.00	1	1
000428321a	E72 Assignment	Franks	Alex	al08d@acu.edu	0.00	0%	0.00	0	0
0004b33167	E72 Assignment	Donaghey	Anna	ai09a@acu.edu	7.92	79%	9.50	1	1

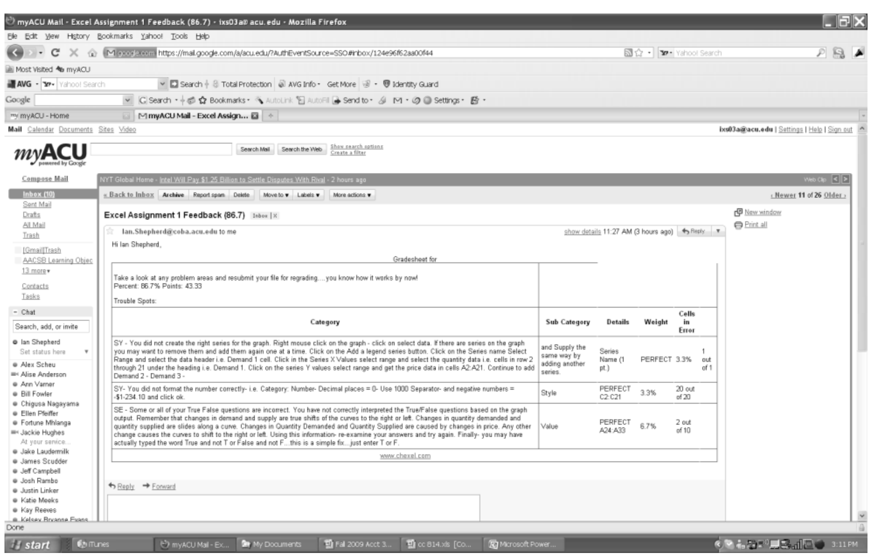
In the prototype desktop system the teacher could (on a scheduled basis) grade all files in the submissions folder. The prototype system checked answers, based on the rules created by the teacher i.e. formulas, formats, ranges, and correct answers. On a typical desktop system, the system graded 200 workbooks in under 2 minutes. The new web based system grades each file instantaneously as the student uploads the file.

Part of the prototype functionality of the system included email notification. After grading, the students were informed via email of their assignment grade and exactly what was wrong. The student email contained feedback directions pertaining to each of the student's deficient areas in the assignment.

As part of the grading process, the prototype created feedback for the teacher that allowed them to focus on those areas within the assignment where the majority of students fail to understand a concept or fail to grasp a technology skill. This enabled early diagnosis of problem areas and helped the teacher clear up confusion and give extra instruction in specific areas. The teacher could address these deficient areas either in class or in a special session with the students.

The web based system grades each student assignment immediately as the assignment is dropped on the web site. Immediate feedback from the web removes time delays in the old desktop system where the student was relying on the teacher to manually run a grading process.

FIGURE 7
STEP 7 – REVIEW, REPAIR, AND RETURN THE ASSIGNMENT



In the prototype desktop system, the student reviewed the feedback, amended the file and returned the assignment to the teacher for re-grading and possible re-submission back to the student with further directed instructions on areas where the student has failed to comply with the assignment instructions. Iterative assignment grading is an option for the teacher.

Iterate Early and Often

One of the most important factors in learning is iteration. Humans learn best in small, iterative steps. Because the prototype tool graded so quickly, the part of an assignment that used to take the most time now took the least amount of time. This enabled the teacher to give feedback "early and often." Rather than accepting homework only once right before a deadline, the prototype system allowed the teacher to accept assignments early and grade them often.

Iterative assignments contributed to learning in an important way. The nature of technology integration was that small errors (syntax) could lead to large penalties (one formula is wrong and all dependent cells thus also wrong) [Hristova 2003]. Although we live in a world in which small errors can certainly lead to large consequences, the creators of the process do not believe this is the best way to teach. On the contrary, the authors believe that allowing iteration on assignments helps the student find syntax errors which have resulted in serious semantic errors. Clearly the syntax must be correct before the semantics can be considered correct. Students cannot speak intelligently about elasticity if the formulas that create data used to understand that concept are incorrect.

Once the syntax is correct, how can we also evaluate semantics? The authors have discovered that by attention to learning outcomes and careful phrasing of questions, teachers can use syntactic markers to communicate semantics. For example, referring to a table with data, one can challenge the student to "use bright green for all cells that show inelastic demand." To get this question right, the correct technology skills must be in place (right formulas - syntax) and the domain concepts must be understood (elasticity - semantics).

By allowing iteration, the student receives feedback on both the "how" and the "what" of the assignment. This feedback is directly related to the skill level and competency of each student. The ability of this system to manage large numbers of students not only allows schools to maintain the economic benefits of larger class sizes, but begins to focus more closely on individual performance and instruction.
A counter argument to allowing iteration is that students must learn how to get it right the first time.
Our experience with under graduates causes us to be more interested in the lower 99% than the top 1%, who are capable of getting it right the first time. There are certainly times to teach that precision is needed right now, but that lesson is not the most important lesson and we believe most students benefit more from a gradual and iterative approach.

THE HYPOTHESIS

The authors propose that ***assignment iteration*** decreases technology errors (those errors that fog the interpretation of economic data - ***syntax errors***), improves technology competence (I can repeatedly generate correct economic formulas), and improves domain knowledge (I understand how to interpret this economic data - ***semantic errors***).

Research was conducted with a regular and online Macroeconomics and Microeconomics courses and data was collected from two self-selecting groups. The first student group chose to use an iterative learning approach. The second student group chose to use single submissions of the required excel assignments. The iterative assignment option was offered to all students. Students who iterated at least once during the semester were counted in the iterative group. No measurement of student motivation was made during the courses. Self-selection and use of iteration might imply a more motivated student.

The students took one of two routes: Iteration or No Iteration. A summary of the differences in each rout appears below.

FIGURE 8
GROUPS – ITERATION AND NO ITERATION

Iteration Group	*No Iteration Group*
Feedback	No Feedback
Multiple Submission	Single Submission
Iteration Improvement Measurements	
*Syntactic - score change	
*Semantic - score change	
Semantic - Post-test Exam Score	Semantic - Post-test Exam Score
Syntactic - Post-test Excel Skill Set	Syntactic - Post-test Excel Skill Set
Student Evaluation of process	Student Evaluation of process

Iterative Feedback and No Feedback

Feedback was conveyed in the prototype system to those who chose to iterate via email. Email feedback error messages were classified as either: syntax (SY) or semantics (SE) based on the requirements of the assignment.

1. Syntax feedback typically dealt with Excel skill issues; such as the student's inability to create a formula. Syntax feedback instructions were designed to specifically guide the student in correcting the syntax error prior to making any interpretation of the data for the assignment.
2. Semantic feedback typically dealt with interpretation issues surrounding the data; such as elasticity ranges. If the data were correct, and the student misinterpreted that data, then instructions were given on where to focus to correct this misinterpretation i.e. inelastic data is less than 1.

In addition to the syntax and semantic feedback, the emails weighted the student errors showing the student where the greatest percent of their grade was missed. This allowed the student to focus on the errors of greatest magnitude, and thus offered the student the greatest opportunity to improve their grades. Careful consideration was given to feedback to ensure that prior dependencies were noted so that cascading errors could be tracked.

Students choosing not to iterate received only the first graded email and chose to take the first and final grade for their assignments.

CURRENT RESEARCH

Goffe and Sosin (1995) note that while the use of computers and the web within the classroom has increased, there is hesitancy for instructors to use computer modeling tools to dynamically test students understanding of economic concepts. Resistance, they maintain, comes in two forms: both instructor and student hesitance in using the new technologies. The difficulty in implementing these new techniques is compounded by two factors: the instructor must redefine modeling assignments to convey the economic concepts, and the student must overcome poor technology skills to be able to use the modeling technique.

Experimentation within the classroom with assignments and models that allow the student to build data and understand relationships helps students improve both their attitudes and understanding of economic concepts (Grimes, Ray 1993). The problem then becomes, how might the instructor "crest the technology wave, increase modeling within the economics course load, and reduce student resistance to learning new technology skills." Goffe and Sosin (Goffe, Sosin 2005) discuss the need to also measure improved performance on the part of the student i.e. is the technological effort worthwhile?

Seven years of data collection and twelve years of program development at Abilene Christian University produced the prototype system (and now the new web based system) that addresses the concerns of instructors wishing to design, implement, and measure the use of technology in the classroom modeling environment. Shepherd and Reeves (Shepherd, Reeves 2006) describe the prototype system.

The instructor distributes model templates to students who complete the economics assignment. The students return the assignments to the instructor. The assignments are graded and feedback sent via email. With the burden of grading removed, assignments that are submitted early can be graded, feedback generated at the individual level, and error information returned to the student for review via email, allowing correction and resubmission by the student. Feedback design is important and requires the instructor to spend time defining the requirements for the assignment. Here it is up to the instructor to define the types of errors i.e. incorrect formula, failure to provide formulas, failure to use the right function, and failure to interpret the data correctly. Additional presentation skills can be developed at the instructor's request to enhance the student's ability to present visibly pleasing data in formats that convey the correct interpretation of the data i.e. graphs, titles, and data formats.

Once problems with feedback are reduced, and the ability to address individual errors is addressed, email (or a web page presentation about the errors) becomes a powerful tool in correcting modeling errors. Iteration now becomes manageable and in fact desirable.

Along with the submission of electronic assignments came the need to step up the students' skills in managing data movement over the web. Experience in using the prototype system showed that strict rules with regard to assignment submission actually enhanced the student's ability to diagnose delivery problems i.e. in the drop box by 11:55 pm on due date. Delivery methods could vary; ftp, Blackboard file move, Explorer copy, Explorer move, Save to from Excel, Save as from Excel, and now upload to a web page. All students became aware that on-time delivery of a correct product had its benefits - a good grade.

Novak etal (Novak etal 1999) first suggested that students would benefit from interactive activities in the classroom accompanied by web based resources that helped the students develop basic economics skills. They defined this technique as JiTT or Just-in Time Teaching. The basis of JiTT is that class activities and homework should encourage outside development by the students, provide quick feedback, and allow the instructor to modify future classes and assignments to address learning deficiencies.

With grading and feedback instantaneous to students, the instructor is able to identify problem areas quickly, refocus either class instruction, and/or redesign future assignments to follow a track that helps the students clarify learning problems. Simkins and Maier (Simkins, Maier 2004) developed an innovative teaching technique in their introduction to economics classes that designed future classes based on question feedback from students. The prototype system can be used both in (where students have access to computers) and outside the classroom to determine exact areas of deficiency. Instructors are presented with weighted errors and can focus attention on correcting errors in semantics or syntax based on full class responses to assignments i.e. 42% of the class cannot identify the inelastic range of this data and 15% cannot correctly create the formula for elasticity.

Research in the computer science area has shown that Web-CAT automatic grading systems help students focus their efforts through graphic representation of the student's relative position within the class allowing the students to iteratively improve their assignments (Edwards etal 2006). They maintain that students need to not only see their problem areas, but that they need to be able to place themselves in positions of comparison to other students on the same assignments. Edwards (2003) maintains that this feedback is also invaluable to the instructor as it helps focus the instructor on areas of deficiency thus allowing a modified JIT teaching approach to resolve areas of deficiency.

Malmi (2004) maintains that "it is often much better to get instantly even simple feedback than to get advanced human feedback many days afterwards, or even worse to get no feedback at all." The purpose of the prototype system was to provide this feedback on a timely basis. Malmi's research also directed further research be done to focus on the types of errors involved by the students that limit their understanding of the course content. The authors support Malmi's request for error tracking through data collection at the error and feedback level. The teacher is now empowered with the ability to assess and analyze error data to adjust teaching methodology.

The ability of the system to categorize errors based on instructor requirements is a major step forward in removing barriers to learning while enhancing student interaction and feedback so as to remove these errors.

Improved Student Scores

Improved Domain Knowledge

A total of 45 students enrolled and completed the courses. Of the 45 students, 39 chose to iterate assignments at least 1 time and 6 chose not to iterate assignments. Comparisons of students who did not iterate and did iterate found that on average students who iterated improved their grades by 23% compared to those who do not iterate.

An independent Samples T-Test was performed on the average final grades for both groups (iterate vs. not-iterate). As shown in Table 1 and 2 below:

TABLE 1
GROUP STATISTICS – AVERAGE GRADES COMPARISON BETWEEN GROUPS

Student Iterated?	N	Mean	Std. Deviation	Std. Error Mean
Y	39	665	95.4936	15.2912
N	6	458	227.2878	92.7898

TABLE 2
INDEPENDENT SAMPLES TEST – AVERAGE FINAL GRADES COMPARISON - BETWEEN GROUPS

	Levene's Test for Equality of Variances		*t-test for Equality of Means*		
	F	*Sig.*	*t*	*df*	*Sig. (2-tailed)*
Equal variances assumed	5.252	0.027	3.979	43	0
Equal variances not assumed			2.201	5.275	0.076

Table 1 displays the mean points for both groups. The group of students that iterated had a mean of 665 points compared to 458 for the non-iterate group. **Table 2** displays the results of a Levene's test for equality of variances. This analysis was conducted due to the large standard deviations associated with each group. Further analysis is being done to identify the source of this large deviation (possible problems include: a student starting the course and not finishing the course – dropping out and not submitting all the work required).

As shown, the Levene's test was significant ($p < .05$), and therefore the Equal variances not assumed t-test must be used. Unfortunately, those results are not statistically significant despite the large difference in the mean points for the two groups.

Average assignment scores for students who did not iterate were 7.5 out of a possible 10 while students who did iterate averaged 9.2 out of a possible 10 for their assignments. The average score and standard deviation per assignment for those students that chose to iterate was calculated. The maximum number of iterations was four. Table 3 below displays the average score per iteration (out of 10).

TABLE 3
AVERAGE SCORE AND STANDARD DEVIATION FOR THOSE WHO ITERATED

Iteration #	*N*	*Mean*	*Std. Deviation*
1	128	6.2184	2.98621
2	127	8.5469	2.0813
3	27	8.9481	2.06568
4	3	9.0567	1.00481
Total	***285***	***7.5445***	***2.7858***

An ANOVA was conducted on the average score per iteration and is displayed in Table 4.

TABLE 4
ANALYSIS OF VARIANCE FOR COMPARISON OF AVERAGE SCORES FOR THOSE WHO ITERATED

Iteration #	*N*	*Mean*	*Std. Deviation*
1	128	6.2184	2.98621
2	127	8.5469	2.0813
3	27	8.9481	2.06568
4	3	9.0567	1.00481
Total	***285***	***7.5445***	***2.7858***

As shown, the ANOVA is statistically significant indicating that average grades improved significantly as students elected to iterate.

Improved Technology Competence

The study grouped the learning of new Excel skills into the first five assignments. No new technical skills were required after assignment five. Visual data groupings imply that by assignment five, the number of times students iterated dropped from three iterations to one iteration.

TABLE 5
ASSIGNMENT COUNT BY ITERATION TYPE GREEN AREA

Assignment Count by Iteration Type													
Assignment Count	*No Iteration*	*Iteration*										*Total*	*Total*
Description	*1*	*1 of 1*	*1 of 2*	*2 of 2*	*1 of 3*	*2 0f 3*	*3 of 3*	*1 of 4*	*2 of 4*	*3 of 4*	*4 of 4*	*Assignments Graded*	*Students*
Assignment 0	6	10	13	13	2	2	2					48	31
Assignment 1	6	7	12	12	7	7	7					58	32
Assignment 2	5	9	14	14	2	2	2					48	30
Assignment 3	7	19	4	4	1	1	1					37	31
Assignment 4	6	14	8	8	2	2	2					42	30
Assignment 5	7	20	3	3	1	1	1					36	31
Assignment 6	4	16	5	5	2	2	2					36	27
Assignment 7	4	15	6	6				1	1	1	1	35	25
Assignment 8	3	19	5	5								32	27
Assignment 9	6	12	10	10	1	1	1	1	1	1	1	45	29
Assignment 10	5	17	6	6	1	1	1					37	29
Assignment 11	5	19	5	5	1	1	1					37	30
Assignment 12	3	15	5	5	1	1	1	1	1	1	1	35	24
Assignment 13	4	12	7	7	4	4	4					42	27
Assignment 14	4	21	1	1								27	26
Assignment 15	3	15	6	6								30	24
Total	*78*	*240*	*110*	*110*	*25*	*25*	*25*	*3*	*3*	*3*	*3*	*625*	*453*

* Note: Light brown shaded area represents all the Excel Skills learned.
* Note: Light green shaded area represents a visual grouping of skill mastery.

Examining the last 10 assignments, we wanted to see if there was a difference in the number of total errors between those who iterated and those who did not. Expectations were that there would be a difference as students who iterated were more likely to resolve errors earlier in the learning process than students who did not iterate. Table 6 shows the results for the comparison of the last 10 assignments. Students who iterated had .23 mean errors compared to .36 mean errors for non-iteration students.

TABLE 6
GROUP STATISTICS – TOTAL

Group Statistics					
Student_iterate		*N*	*Mean*	*Std. Deviation*	*Std. Error Mean*
Top10Err	*Y*	571	0.23	0.842	0.035
	N	80	0.36	0.984	0.11

Independent Samples Test										
		Equality of Variances		*t-test for Equality of Means*						
									95% Confidence Interval of the Difference	
		F	*Sig.*	*t*	*df*	*Sig. (2-tailed)*	*Mean Difference*	*Std. Error Difference*	*Lower*	*Upper*
Top10Err	*Equal variances assumed*	4.468	0.035	-1.33	649	0.184	-0.137	0.103	-0.338	0.065
	Equal variances not assumed			-1.182	95.89	0.24	-0.137	0.116	-0.366	0.093

Given that the equal variances assumed results were significant, we must use equal variances not assumed results. These indicated that there was no significant difference between the mean results (however, differences were indicated).

Further breakdown of the error analysis allowed us to compare just syntax errors over the last 10 assignments. Expectations were that we would see a significant difference in syntax errors between the two groups. Those who iterated would be expected to have a lower occurrence of syntax errors.

TABLE 7
COMPARISON OF SYNTAX ERRORS ON LAST 10 ASSIGNMENTS

Group Statistics

Student_iterate		N	Mean	Std. Deviation	Std. Error Mean
Nsynerr	Y	571	1.37	1.779	0.074
	N	80	1.85	1.994	0.223

Independent Samples Test

		Lavene's Test for		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Top10Err	Equal variances assumed	1.152	0.284	-2.236	649	0.026	-0.482	0.216	-0.906	-0.059
	Equal variances not assumed			-2.051	97.438	0.043	-0.482	0.235	-0.949	-0.016

Group Statistics

	student_iterate	N	Mean	Std. Deviation	Std. Error Mean
number of syntax 1 occurences	Y	571	.45	.498	.021
	N	80	.59	.495	.055
number of syntax 2 occurences	Y	571	.24	.428	.018
	N	80	.33	.471	.053
number of syntax 3 occurences	Y	571	.16	.363	.015
	N	80	.19	.393	.044
number of syntax 4 occurences	Y	571	.17	.376	.016
	N	80	.18	.382	.043
number of syntax 5 occurences	Y	571	.12	.330	.014
	N	80	.21	.412	.046
number of syntax 6 occurences	Y	571	.05	.212	.009
	N	80	.09	.284	.032
number of syntax 7occurences	Y	571	.04	.205	.009
	N	80	.11	.318	.036
number of syntax 8 occurences	Y	571	.05	.223	.009
	N	80	.06	.244	.027
number of syntax 9 occurences	Y	571	.03	.180	.008
	N	80	.04	.191	.021
number of syntax 10 occurences	Y	571	.02	.138	.006
	N	80	.04	.191	.021
number of syntax 11 occurences	Y	571	.01	.102	.004
	N	80	.01	.112	.013
number of syntax 12 occurences	Y	571	.01	.118	.005
	N	80	.01	.112	.013
number of syntax 13 occurences	Y	571	.00	.042	.002
	N	80	.00	.000	.000
number of syntax 14 occurences	Y	571	.00	.042	.002
	N	80	.00	.000	.000
number of syntax 15 occurences	Y	571	.00	.042	.002
	N	80	.00	.000	.000

Table 8 shows that there is a significant difference between the two groups with regard to syntax errors. Iteration here appears to have improved the student's ability to avoid Excel errors. Table 8 compares each individual assignment for syntax errors.

TABLE 8
INDEPENDENT COMPARISON OF SYNTAX ERRORS BY ASSIGNMENT

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
number of syntax 1 occurences	Equal variances assumed	2.427	.120	-2.313	649	.021	-.137	.059	-.254	-.021
	Equal variances not assumed			-2.322	102.663	.022	-.137	.059	-.255	-.020
number of syntax 2 occurences	Equal variances assumed	7.893	.005	-1.608	649	.108	-.083	.052	-.185	.018
	Equal variances not assumed			-1.497	98.171	.138	-.083	.056	-.194	.027
number of syntax 3 occurences	Equal variances assumed	1.948	.163	-.722	649	.470	-.032	.044	-.118	.054
	Equal variances not assumed			-.681	98.848	.498	-.032	.046	-.124	.061
number of syntax 4 occurences	Equal variances assumed	.051	.821	-.114	649	.909	-.005	.045	-.093	.083
	Equal variances not assumed			-.112	101.579	.911	-.005	.046	-.095	.085
number of syntax 5 occurences	Equal variances assumed	15.753	.000	-2.164	649	.031	-.088	.041	-.168	-.008
	Equal variances not assumed			-1.834	93.785	.070	-.088	.048	-.184	.007
number of syntax 6 occurences	Equal variances assumed	8.670	.003	-1.514	649	.130	-.040	.027	-.092	.012
	Equal variances not assumed			-1.218	91.761	.226	-.040	.033	-.106	.025
number of syntax 7occurences	Equal variances assumed	24.790	.000	-2.597	649	.010	-.069	.026	-.121	-.017
	Equal variances not assumed			-1.879	88.408	.064	-.069	.037	-.141	.004
number of syntax 8 occurences	Equal variances assumed	.537	.464	-.369	649	.712	-.010	.027	-.063	.043
	Equal variances not assumed			-.346	98.510	.730	-.010	.029	-.067	.047
number of syntax 9 occurences	Equal variances assumed	.152	.697	-.196	649	.845	-.004	.022	-.047	.038
	Equal variances not assumed			-.186	99.511	.852	-.004	.023	-.049	.041
number of syntax 10 occurences	Equal variances assumed	4.331	.038	-1.052	649	.293	-.018	.017	-.052	.016
	Equal variances not assumed			-.824	90.812	.412	-.018	.022	-.062	.026
number of syntax 11 occurences	Equal variances assumed	.104	.747	-.162	649	.872	-.002	.012	-.026	.022
	Equal variances not assumed			-.151	98.336	.880	-.002	.013	-.028	.024
number of syntax 12 occurences	Equal variances assumed	.047	.829	.108	649	.914	.002	.014	-.026	.029
	Equal variances not assumed			.112	105.057	.911	.002	.013	-.025	.028
number of syntax 13 occurences	Equal variances assumed	.562	.454	.374	649	.708	.002	.005	-.007	.011
	Equal variances not assumed			1.000	570.000	.318	.002	.002	-.002	.005
number of syntax 14 occurences	Equal variances assumed	.562	.454	.374	649	.708	.002	.005	-.007	.011
	Equal variances not assumed			1.000	570.000	.318	.002	.002	-.002	.005
number of syntax 15 occurences	Equal variances assumed	.562	.454	.374	649	.708	.002	.005	-.007	.011
	Equal variances not assumed			1.000	570.000	.318	.002	.002	-.002	.005

SUMMARY

- Student Grades improved if they used iteration through the prototype system.
- Iteration improved the student's ability to avoid Excel errors and thus remove technology barriers to learning economic concepts.

CONCLUSION

Tools are now available that provide individual feedback related to the skill level and competency of each student. The focused use of the system provides feedback that enhances learning through iteration. The ability of this tool to manage large classes allows us to maintain the economic benefits of larger class sizes, but begin to focus more closely on individual performance and instruction. The successful application of this tool enhances the technology skills required for the business world, and the subject knowledge skills required to successfully fulfill course content requirements.

Finally, this tool facilitates a change of focus in instructional methods that leads to an improved quality of teaching experience. As professors become comfortable with this tool they are able to focus on what they need to teach students, rather than the drudgery of grading.

REFERENCES

Chamilliard, A.T., and Merkle, Laurence D. (2002), Management Challenges in a Large Introductory Computer Science Course, Proceedings of the 32nd SIGCSE technical symposium on Computer science education, SIGCSE'02, pp. 252-255, Feb 27, 2002

Stephen H., Perez-Quinines, Manuel A., Phillips, Mathew, and RajKumar, Johnny, (2006) Graphing Performance on Programming Assignments to Improve Student Understanding, 9th International Conference on Engineering Education, Session R1H. San Juan, PR.

Edwards, Stephen H. (2003), Teaching Software Testing: Automatic Grading Meets Test-First Coding. OOPSLA 03, October 26-30, 2003, Anaheim, California, USA.

Goffe, William L., Sosin, Kim (2005). Teaching with Technology: May You Live in Interesting Times. The Journal of Economic Education, Vol. 36, No. 3, pp. 278-291.

Grimes, P. W., and M. A. Ray (1993). Economics: Microcomputers in the College Classroom - A review of the academic literature. Social Science Computer Review 11 (Winter): 452-63

Histova, Maria, Misra, Ananya, Rutter Megan, and Mercury, Rebecca (2003), Identifying and Correcting Java Programming Errors for Introductory Computer Science Students, Proceedings of the 33rd SIGCSE technical symposium on Computer science education, SIGCSE '03, Reno, Nevada, pp. 153-156, Feb 19, 2003

Horgan, Barbara (1998), "Transforming Higher Education Using Information Technology: First Steps" *The Technology Source*, January 1998. Available online at http://ts.mivu.org/default.asp?show=article&id=1034.

Kay, David G (1998). Large Introductory Computer Science Classes: Strategies for Effective Course Management, Technical Symposium on Computer Science Education, Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education, Atlanta, Georgia, pp. 131-134, 1998

Malmi, Lauri, and Korhonen, Ari (2004). Automatic Feedback and Resubmissions as Learning Aid. Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)

Meiselwitz, Gabriele (2002), Using the Web to maintain the benefits of small class instruction in large classes, Journal of Computing Sciences in Colleges, Volume 17, Issue 3, pp. 141-148, 2002

Novak, G., Patterson. E., Gavrin. A., & Christian, W (1999). Just-in-time teaching: Blending active learning with web technology. Upper Saddle River, NJ; Prentice Hall.

Piaget, J (1969). The Mechanisms of Perception, New York; Routledge Kegan Paul, 1969

Shepherd, I. & Reeves, B (2006). How to Structure and Evaluate Information Technology Assignments. *Association of Business Information Systems, Federation of Business Disciplines, 1 (1)*, 53 - 56.

Shepherd, Ian J. (2005), Faculty Assignment Load Analysis, College of Business Administration, Abilene Christian University, Technical Report COBA2005031, 2005.

Simkins, S. & Maier, M (2004). Using Just-in-Time Teaching Techniques in the Principles of Economics Course. Social Science Computer Review 2004; 22; 444

Suchman, L (1987). Plans and Situated Action, Cambridge University Press

APPENDIX

ASSIGNMENT DATA

Subject	*Assignment #*	*Total*	*Items Checked*
Macro	0	235	37,130
	1	334	39,950
	2	305	44,415
	3	219	2,350
	4	212	6,580
	5	221	12,455
	6	228	2,350
	7	277	12,690
	8	224	70,030
	9	272	11,985
	10	230	9,870
	11	200	5,875
	12	238	7,990
	13	271	9,400
	14	184	19,035
	15	243	3,055
	16	114	4,700
	17	76	7,050
Macro Total		***4,083***	***306,910***
Micro	0	84	13,272
	1	124	14,280
	2	105	15,876
	3	78	840
	4	82	2,352
	5	80	4,452
	6	89	840
	7	107	4,536
	8	76	5,712
	9	107	20,748
	10	105	14,784
	11	84	25,032
	12	96	27,636
	13	84	1,848
	14	73	6,804
	15	96	4,704
	16	40	5,376
	17	39	4,536
Micro Total		***1,549***	***173,628***
Grand Total		***5,632***	***480,538***