Real Time Engineering Systems Course; Methods for Self-Assessment and Evaluation

Ted Sarma, Massood Z. Atashbar, Hossein Mousavinezhad

Department of Electrical and Computer Engineering
Western Michigan University
Kalamazoo, Michigan, 49008

Abstract

University Computer Engineering programs continue to be a popular draw for students. Still, since they are relatively new, their defining curricula continue to evolve. Traditional courses such as digital logic, and digital design, microcontrollers, computer interfacing and computer architecture are mainstays, but there continues to be many holes to fill. Part of the problem is that Computer Engineering (CE) is still considered to be an interface between Electrical Engineering (EE) and Computer Science (CS). Electrical Engineering, where it is usually housed, embraces the notion that computer hardware is fundamental to the discipline while Computer Science views computer software as the defining entity. The truth is that both are correct and Computer Engineering students need to understand both disciplines equally well. At the same time, this understanding needs to go beyond simply knowing about EE and CS. Students must be able to apply the principles of high-level system analysis and design techniques to electrical engineering applications.

It is the authors' belief that one of the most common areas between EE and CS encompasses digital data acquisition, signal processing, communication and control. Coincidentally, these turn out to be some of industry's major needs as well. At the same time, students need to be exposed to a reasonable amount of high-level software engineering practices that are engineering based. However, there is no way that an undergraduate CE program can accommodate each of these courses in an already crowded curriculum. The solution to this problem, that has been implemented at Western Michigan University (WMU), is to create a junior level course that teaches high-level software engineering best practices using Visual Basic that is applied to data acquisition, signal processing and network communications. In addition, the students are required to maintain assignment logs providing a closed-loop feedback mechanism for continuous improvement in the quality of the course and their learning experience. This course has been highly successful in that students not only learn a great deal of information but also gain experience in applications that will be useful in further course work and senior projects as well as their future careers.

Introduction

Many educational institutions utilize single assessment methodologies (i.e. university- wide class surveys) to gage the students' satisfaction of collective and individual course offerings. These assessments tend to be generalized and may not provide appropriate feedback regarding specific course materials. We have implemented several assessment methodologies, which provide continuous feedback during the course session.

The foundational course of this paper was first documented by Atashbar [1]. "Engineering of Real-Time Systems" was introduced as a new junior level engineering course in the Fall of 1999 at Western Michigan University. This course addressed the education of EE and CE in the areas of digital data acquisition, signal processing, communications and controls. Student feedback in the form of standard student reviews indicated significant satisfaction with the course structure and materials.

In the Fall 2002 semester, we enhanced the course by adding some basic Software Engineering best practices. This included an initial student knowledge assessment questionnaire and an engineering log sheet to be completed with each assignment. The assessment questionnaire provided valuable feedback to the instructor as to the students' level of knowledge about Software Engineering concepts. The log introduces the students to good engineering practices by requiring them to document their software development process. The log also included a student assessment of the assignment. The instructor is able to monitor the size and complexity of each assignment and the students' satisfaction with course materials.

Initial Assessment

The level of student knowledge and skills as they relate to the course materials should be determined as early as possible. To this end, we have devised a simple questionnaire, which makes a non-threatening determination of these levels. Three primary areas were covered for the "Engineering of Real-Time Systems" course, namely:

1. Software programming and languages experience
2. Computer operating systems familiarity
3. Software Engineering terminology

Results from the questionnaires have indicated that most EE, CE students do some familiarity with certain programming languages and operating systems but not necessarily with the programming environment used in the course. It was surprising that most students were not familiar with many of the Software Engineering terminologies. The Visual Basic development environment and language were covered in reasonable detail so as to enable the students to complete each assignment. The course materials were adjusted to include an introduction to the basic concepts behind the Software Engineering terminology. A sample questionnaire is shown in Figure 1.

ECE **Questionnaire**

Name: _____ ID:_____

Programming Language Familiarity (check the appropriate level)

	None	Novice	Moderate	Experienced	Guru
Assembly Language	☐	☐	☐	☐	☐
Basic	☐	☐	☐	☐	☐
Microsoft Visual Basic	☐	☐	☐	☐	☐
C++	☐	☐	☐	☐	☐
Microsoft Visual C++	☐	☐	☐	☐	☐
Other, Specify:_____	☐	☐	☐	☐	☐

Operating Systems Familiarity (check the appropriate level)

	None	Novice	Moderate	Experienced	Guru
Microsoft Windows 9x	☐	☐	☐	☐	☐
Microsoft Windows NT	☐	☐	☐	☐	☐
Linux	☐	☐	☐	☐	☐
Other, Specify:_____	☐	☐	☐	☐	☐

Software Engineering Terms (check the appropriate level)

	Never Heard of it	Heard of it	Taken a course	Actually applied	Guru
Capability Maturity Model (CMM)	☐	☐	☐	☐	☐
Personal Software Process (PSP)	☐	☐	☐	☐	☐
Total Quality Management (TQM)	☐	☐	☐	☐	☐
Constructive Cost Model (CoCoMo)	☐	☐	☐	☐	☐
Unified Modeling Language (UML)	☐	☐	☐	☐	☐
Software Life Cycle	☐	☐	☐	☐	☐
Configuration Management	☐	☐	☐	☐	☐
Software Reusability	☐	☐	☐	☐	☐

Figure 1: Questionnaire

Assignment Log

Engineers and researchers generally maintain notebooks recording their activities. The notebooks and logs can subsequently be reviewed to validate that certain procedures were followed. Analysis of the appropriate recorded measures, such as task times, can be used to facilitate process improvement. The notebook can also be used in assessing the engineer's performance.

The assignment log was designed to encourage the students to document their development processes and to provide the instructor with relevant course measures. The inclusion of specific code size metrics and assessment methods provided valuable course feedback. Subsequent course materials and assignments were improved using this information.

The assignment log is based on the materials described by Humphrey [2] as part of the Personal Software Process and the Strengths/Improvements/Insights (SII) assessment method articulated by Apple [3]. The assignment log required the students to record the following information for any given assignment:

1. Software development activities. Each individual activity required the following:
    a. A sequential identifier
    b. Date
    c. Start time
    d. End time
    e. Time lost to unexpected interruptions
    f. Activity type. The activities were broadly categorized into
        i. Preparation
        ii. Reading
        iii. Code design
        iv. Coding
        v. Testing
        vi. Defect removal
        vii. Documentation
    g. A brief description

2. Defect documentation. Each defect (syntax errors, incorrect coding logic, unexpected application behaviors) required the following information:
    a. The identifier of the activity during which the defect was introduced into the application.
    b. The identifier of the activity when the defect was actually discovered.
    c. The defect type
    d. A brief description

3. Code metrics. The code metrics included:
    a. Total number of lines of code (LOC), including comments
    b. Number of lines of comments
    c. Total number of subroutines and functions coded
    d. The number of code files in the assignment

4. Assignment assessment. This required the student to provide one or more details in each of the assessment areas:
    a. Strengths
    b. Improvements
    c. Insights

This course was designed so that the student would be able to use Visual Basic to create virtual instruments and be able to implement concepts such as data acquisition (including sampling, Nyquist criteria, Fourier Transforms and feature extraction), signal conditioning, sensors, serial and parallel asynchronous communications and control. Each topic was demonstrated at the hardware level and formed the basis of a software assignment. Assignments are listed along with a short description as follows:

1. *Demonstration Program*: A Visual Basic version of the "Hello World" program in which a variety of VB components are demonstrated.

2. *Ohm's Law:* A program emphasizing error trapping and conditional results.
3. *Descriptive Statistics*: Reading/writing files, creating statistical tables, histograms and traditional statistics.
4. *Oscilloscope*: Sampling a signal using software interrupts and displaying the results graphically as a virtual oscilloscope in real time.
5. *Spectral Analyzer*: Incorporating an FFT module with the Oscilloscope program to create Bode plots in real time.
6. *Data Acquisition*: Use of a third part (National Instruments) activeX oscilloscope component along with a DAQ card for data acquisition.
7. *Serial Communication*: Creation of an asynchronous serial communication protocol to acquire data from the serial port.

A typical assignment log document is shown in the Figure 2. Additional student incentive was provided by designating between 15 to 25 percent of each assignment's marks to the submission of a properly completed log.

ECE - Assignment Log Assignment #: _____

Name: _____ ID: _____

#	Date	Start	End	Int	time	Activity	Comments
1							
2							
3							
4							
5							
6							
7							
8							
9							
A							
B							
C							
D							
E							
F							
G							

#Lines of Code: ____ #Lines of Comments: ____ # of Subs/Functions: ____ # of Files: ____

Defect(s)			
Activity		Type	Description
Injected	Removed		

Defect Types
1 – Documentation
2 – Syntax
3 – Build
4 – Assignment
5 – Interface
6 – Checking
7 – Data
8 – Function
9 – System
A – Environment

Assignment Assessment	
Strength(s)	
Improvement(s)	
Insight(s)	

Figure 2: Assignment Log

Results

Data collected from several questionnaires indicated that most students do not have a significant knowledge of Software Engineering terminology and therefore would not be versed in the associated practices. As a result, a brief introduction to basic Software Engineering practices such as the Capability Maturity Model and Personal Software Processes were included in the course curriculum. This provides the background and framework for the assignment logs.

The assignment logs provide significant feedback to the instructor. We were able to determine the approximate time spent on each assignment. Also, the size and complexity of the assignments were measured. The assessment portion of the log provided information on the strengths and weakness in the assignments and subsequently, an indication of which areas would require more attention. Data collected from a class of 26 students during the Fall semester of 2002 are presented in Tables 1, 2, 3, and 4.

Table 1 shows the typical average time consumed by the students in completing individual assignments. As it can be seen the median of the consumed time varies as the complexity of the assignment. Although large standard deviations have been registered, these are attributed to the fact that some students have little or weak programming backgrounds. However, Table 2 indicates that students believe the designed assignments were a good way to learn about the concepts and implementation of virtual instruments. Table 3 itemizes some student suggestions for areas of improvement. The common consensus has been that the assignments should contain more detailed specifications for the assignments' tasks. Table 4 shows the insights collected by the student assessments. The last assignment, which was the concluding assignment for the semester, indicated that the students have a greater appreciation of automation packages such National Instruments' LabView™, MatLAB™, MathCAD, etc. They also realized that developing such packages requires training, hard work, and a true understanding of software and hardware.

Table 1: Assignment Completion Time and Size

#	Assignment Description	Typical Time			Typical Code Size (code & comments) (LOC; comments)
		Average	Std Dev	Median	
1	Simple User interface	211	144	188	88; 30
2	Ohm's Law	730	436	540	446; 60
3	Descriptive Statistics	423	248	372	235; 40
4	Oscilloscope	243	174	177	187; 33
5	Spectral Analysis	438	323	333	178; 30
6	Data Acquisition	434	152	520	224; 42
7	Serial Communications	282	120	272	254; 33

Table 2: Student Assignment Assessments – Strengths

#	Comment
1	"Quick learning and easy to use Visual Basic" "Good easy introduction, helped me remember a lot of the stuff I haven't done in a few years in VB." "Easy to design interface, codes are simple and short."
2	"Uses activity related to major (circuit analysis)" "Makes one read through the book exhaustively in order to code and under stand built in functions. Helps develop insight and intuition for coding in VB" "The assignment was not only a good example of practical work, but also a showed me how to play around with combos, list boxes, validation, conversions etc."
3	"Lots of different controls used, tabs, flex grid, charts, file input, etc" "Good step up from the last assignment"
4	"We discussed the program and the sample on the website thoroughly in class, so that was helpful for the coding of this assignment." "I thought that the directions were not clear in terms of what we were supposed to get from the user. Also, Sampling Rate is a frequency, which should be in Hertz, not milliseconds. So that was confusing at first as well." "Good to work with frequency and sampling issues."
5	"Good way to learn about coding and charting FFTs" "Gives an insight into using Dynamic Link Libraries and working simultaneously with Visual C++"
6	"Got to know how the NI Data Acquisition card (PC-LPM – 16) works. Nice way to get the students to learn how to make hardware interact with the software simulation." "Team worked well together"
7	"The assignment was fairly easy to understand and also easy to find information about." "Good for simulating computer-peripheral activity type thinking, coding habits" "Learn how to send and receive bits in ASCII through COM port using visual basic" "Ability to use ports, very glad to get some hands on experience with ports and communications"

Table 3: Student Assignment Assessments – Improvements

#	Comment
1	"Need more definition. Don't just say make a program to use these button. What is print supposed to do? What type of capitalization would you like in the display area?" "posting assignments as html docs or another open form would be appreciated. If you could" "Not sure on whether the checkboxes and radio buttons should be cleared when CLEAR is clicked, otherwise ok"
2	"More specifics about appropriate input / output" "Assignment to begin with raises questions as to the combinations for input…but then consultation with lecturer clears that up. Assignment could have been more specific, explanatory" "Much more difficult to do than expected. Hard to understand" "It seemed like there was a big jump in the extent of code writing and knowledge needed between the first and second assignments."
3	"VB can make a novice programmer look pretty advanced" "Although creative this was really hard to program. This should be later on in the class. This is a rather complicated program for it to be the third one." "The functions and workings of some controls still not clear"

4	"Better way to graph function so graph doesn't flash during program???" "Could add multiple signals on the same graph." "No guidelines for user interface appearance, tabs/tabbing order"
5	"I am sure I am the only one but I feel the assignment was not clear. I often found myself just giving up because the example was hard to follow on my little computer." "for future assignments I use the sound card to input the data so the data can be real. There is a web site I found that was good about that it was http://www.fullspectrum.com/" "Be more specific about the requirements."
6	"We could carry out data acquisition and the display/operations at the same time (or with a very slight delay) to pose a true real time environment" "The assignment could have at least provided us with more a clearer picture on our goals." "More opportunities to use the lab"
7	"Have the class in the lab. I know that the programs are to be written on our own but I think if we could see how it was supposed to look compared to our own output we could then ask you what is the difference and maybe get some help." "We needed to be able to get into the lab to test this program, but were unable to find a time because of the need to work around lab hours from our other classes that also had projects due this week"

Table 4: Student Assignment Assessments – Insights

#	Comment
1	"good starting assignment, makes you familiar with the ui if you aren't already" "Visual Basic may not be that difficult as I thought." "This form does not take to account the way I program. I write programs then test little modules so my debug time becomes integrated. If I had to write for every time I made a little section of code then tested it I would spend more time on the assignment log then on the coding itself" "There is a lot of documentation involved in software development"
2	"Was really tough" "Learnt more than a few things… function handling,…msg box messages etc…tough assignment" "There will be an assumed level of knowledge and the book is to cover the rest of the material"
3	"This assignment was straight forward and fairly easy. The assignment was fun to do." "This assignment was pretty straightforward and easy to program, not many tricky language things, just the file input" "Learning to program in a new language can be stressful." "I really really really hate how vb beeps you if you dont complete an if statement." "I'm getting better with VB. I don't have to spend quite as much time thumbing through references and examples as I did for previous assignments. But having C++ burned into my brain still tends to slow things down at times. It makes for a lot of syntax errors."
4	"Choice of frequency and interval size make quite a difference in the appearance of the sin wave" "Working environments rarely give guidelines for user interface appearance."
5	"FFTs are cool!" "Working with charts is easy and is fun. The same thing takes longer time in Matlab. Reusability of Code can be a great thing." "I liked doing this program. It was very helpful."
6	"We find the assignment this time round to be very confusing. The first is the use of the NI components which are new to us. Another would be that the NI components are not up-to-date as well. Plus the limited usage of room 3059 also hindered us from completing the assignment on time." "Each of the project team members learned by looking at others code"

7	"Thought it may seem easy to complete , working with outside instruments is hard. Training is everything to complete such tasks." "Good intro to serial communications"

Conclusions

The questionnaire provided a measure of the students' knowledge of Software related practices and subsequently the justification for the introduction of more formal Software Engineering methodologies. The assignment logs have proven to be an extremely useful mechanism for monitoring student progress, effort and satisfaction. The time taken to complete particular assignments indicate that certain assignments did require additional support materials. At the same time, the students did express a great personal satisfaction in completing the more challenging problems. The introduction of formal Computer Engineering best practices and assessments has enhanced the student experience while at the same time providing valuable feedback to the instructor in strengths, improvements and insights. Based on the assessment outcomes, this course is continually updated and improved to reflect new technologies in this dynamically changing areas and to incorporate student feedback.

Bibliography

1. Atashbar M.Z., Mousavinezhad H., and Ogunleye H., "A New Course for Electrical and Computer Engineering Majors:Engineering Real Time Systems", American Society of Engineering Education, Spring 2001 Conference, Montreal, Canada, June 16-19, (2002).

2. Humphrey W. S., Introduction to the Personal Software Process, 1997, Addison Wesley.

3. Apple D., Krumsieg K., Teaching Institute Handbook, 2002, Pacific Crest.

Biographies

Ted Sarma is an Assistant Professor in the Electrical and Computer Engineering Department at Western Michigan University. Professor Sarma's research interests include software engineering, digital electronics and engineering education. Dr. Sarma is an active member of the American Society of Engineering Education (ASEE) and a long-standing member of Institute of Electrical and Electronics Engineers (IEEE). He may be contacted at: ted.sarma@wmich.edu.

MASSOOD ATASHBAR is an Assistant Professor in the Electrical and Computer Engineering Department. Professor Atashbar's research interests include real time system engineering, physical and chemical microsensors development, digital electronics, and engineering education. He has published more than 50 articles in the area of physical and chemical sensors in refereed journals and refereed conference proceedings. Dr. Atashbar is a Senior member of IEEE, and an active member of the following professional institutes and scientific societies:; ASEE, The International Society of Optical Engineering (SPIE). He may be contacted at: massood.atashbar@wmich.edu.

Hossein Mousavinezhad is Professor and chair, Department of Electrical and Computer Engineering. He is an active member of IEEE and ASEE having chaired sessions in national and regional conferences. He is IEEE Region 4 Educational Activities Chair and member of the ASEE North Central Section Executive Board. He is the ECE Program Chair of the 2002 ASEE Annual Conference, Montreal, Quebec, June 16-19. Professor Mousavinezhad is also a Senior Member of IEEE and Chair of the West Michigan Section, he has been a reviewer for IEEE Transactions including the Transactions on Education. His teaching and research interests include digital signal processing (DSP) and Bioelectromagnetics. He may be contacted at: h.mousavinezhad@wmich.edu.