FAST learning: Follow Accomplishments of Senior Teams

Dr. Fernando Garcia Gonzalez, Florida Golf Coast University

Dr. Fernando Gonzalez joined FGCU as an Assistant Professor in the Software Engineering Program in the fall of 2013. Previously he has worked at Texas A&M International University in Laredo, Texas, the U.S. Department of Energy at Los Alamos National Laboratory in Los Alamos, New Mexico and at the University of Central Florida in Orlando, Florida. Dr. Gonzalez graduated from the University of Illinois in 1997 with a Ph.D. in Electrical Engineering. He received his Master’s degree in Electrical Engineering and his Bachelor’s degree in Computer Science from Florida International University in 1992 and 1989. Dr. Gonzalez research interest includes the intelligent control of large scale autonomous systems, autonomous vehicles, discrete-event modeling and simulation and human signature verification.

Dr. Janusz Zalewski, Florida Gulf Coast University

Janusz Zalewski, Ph.D., is a professor of computer science and software engineering at Florida Gulf Coast University. Prior to an academic appointment, he worked for various nuclear research institutions, including the Data Acquisition Group of Superconducting Super Collider and Computer Safety and Reliability Center at Lawrence Livermore National Laboratory. He also worked on projects and consulted for a number of private companies, including Lockheed Martin, Harris, and Boeing. Zalewski served as a chairman of the International Federation for Information Processing Working Group 5.4 on Industrial Software Quality, and of an International Federation of Automatic Control Technical Committee on Safety of Computer Control Systems. His major research interests include safety related, real-time embedded and cyberphysical computer systems, and computing education.
Retention and Recruitment using FAST - Follow Accomplishments of Senior Teams

Abstract—A new didactic method is proposed and described, called FAST, which is an acronym from Follow Accomplishments of Senior Teams, to attract to the discipline students in low level courses and reduce attrition rates. In essence, the method relies on bringing software engineering student teams from senior project courses, who have accomplished some significant results in their classes, to demonstrate and showcase their projects in introductory courses in software engineering and in other STEM disciplines. Students in lower level courses, with assistance of instructor, then analyze the projects specifications, designs, and implementations, and find out about the principles and specific details of software development on a real case study, which is available at hand. Then, depending on each project’s scope, an instructor in a lower level course may choose one of the techniques, such as a demo, exercise, assignment, or even experiment, to enforce learning and motivate the students to increase their chances of staying in the degree program or even switching to the software engineering program from other majors. Typical software projects involved in the first edition of FAST learning were on robotics, wireless sensor networks, microcontrollers, data acquisition and control, and others. These activities definitely engaged students in lower level courses and caused significant excitement about prospects of learning in higher level courses and pursuing careers in software engineering.

Keywords—software engineering education; engineering pedagogy; project-based learning; teamwork;

1.0 Introduction

It is widely known and acknowledged that there are significant problems with attracting students to the STEM (Science, Technology, Engineering and Math) disciplines. As stated in the recent report from the U.S. Department of Education\textsuperscript{1}: “A total of 48 percent of bachelor’s degree students and 69 percent of associate’s degree students who entered STEM fields between 2003 and 2009 had left these fields by spring 2009. Roughly one-half of these leavers switched their major to a non-STEM field, and the rest of them left STEM fields by exiting college before earning a degree or certificate.”

Engineering programs are no exception. In a study by Santiago and Hensel\textsuperscript{2}, the following major reasons are listed for students dropping the engineering programs:

- 22.5% - engineering majors offered do not match my interests.
- 18.2% - do not think I can succeed in Engineering
- 18.0% - I am in academic difficult
- 13.6% - Too much effort required when I am uncertain about what I want to do.

Various approaches have been used to cure the problem, with mixed results, however. Santiago and Hensel\textsuperscript{2} list the following methods enacted in just one university:

This material is based upon work supported by the National Science Foundation under DUE-WIDER Grant No. 1347640. Additional support is acknowledged from NASA through the UCF’s Florida Space Grant Consortium, Award No. NNX10AM01H. Findings and views expressed herein are not necessarily those of the funding agencies.
Computer science, software engineering and computer programming programs also suffer from this phenomenon. As Santiago and Hensel state, “students are switching from engineering into a non-engineering major due to a lack of interest in engineering programs.”

Our own observations confirm that the freshmen year is particularly difficult for most students. At this point in their education they are deciding what major to pursue, discovering their strengths and weaknesses and wondering what their future will be like in their chosen major or the major they are considering. At Florida Gulf Coast University we offer an introductory course, Introduction to Computer Science that is a brief introduction to many of the topics they will study as part of their Software Engineering course work. This presents a challenge to faculty as well.

Our Software Engineering program, like many programs at smaller universities, desires to recruit and retain as many of these students as possible while simultaneously filter out the students with a low probability of success. This is generally accomplished by carefully selecting a level of difficulty that will hopefully separate the students that can succeed from those that will benefit from choosing a different major. Unfortunately, we lose many students that have the capability to succeed but who are not motivated or perhaps are intimidated by the program. The way the professor interacts with the students and poorly taught courses are also factors as well but that is not the focus of this work.

The introductory courses is where students decide if to major in a STEM discipline or leave and is based on their experience in these courses. Unfortunately many of the students who decide to leave the STEM fields have academic abilities equal to those who stay. Our focus is with our freshmen Introduction to Computer Science course which is the course most likely to influence to remain to leave STEM.

The goal of this work is to investigate what influence project presentations and visits from senior students can have on the retention rate for students in this introductory course. We are experimenting with new approaches aimed at removing some of the anxiety associated with unknowns of their future educational experience, and this paper discusses one particular method named FAST Learning. This work is considered work-in-progress and full assessments have not yet been realized. Results of an open-ended survey are presented at the end of the paper and shows promising results. An analysis of the retention rate for students who were in this FAST program versus those who were not will be performed after the end of this semester so that the students have one full semester after completion of the FAST program. Fortunately we offered two sections of this course and only one was involved in the FAST program. However these courses also had different instructors and this will influence the outcomes of the study since the instructor has a large influence on the students retention.

2.0 Outline of FAST Learning Method

The FAST Learning (Follow Accomplishments of Senior Teams) program aims to accomplish two goals. The first is to recruit and retain capable freshmen students from our
introductory courses (Introduction to Computer Science, and to some extent, Introduction to Programming, and Programming Methodology) and the second is to remove the anxiety the students have, stemming from the unknowns of their future educational experiences. The program works by showcasing the accomplishments of our senior students to our freshmen in the Introduction to Computer Science and to have a question and answer session between the senior presenters and the freshmen students.

We invite the senior students from different senior level courses, such as Senior Software Engineering Project, our capstone course, and high-level elective courses, for example, Data Acquisition and Control. Each of these courses is project based, which allows the senior students to present their accomplishments to the freshmen. The presentation sessions are normally divided into two sections. The first is a presentation of all the student projects and the second is a panel discussion where the freshmen ask questions to the senior students. The seniors demonstrate their projects to the freshmen in an interactive session moderated by Instructor. After the session, both freshmen and seniors provide a brief written write-up summarizing their observations and potential benefits the freshmen are getting from the demo session.

2.1 Sample Projects

Projects for the first round of presentations have been divided into two categories: those involving programming in C/C++ and Java from high-level software engineering courses, and those involving graphical programming in LabVIEW from a Data Acquisition and Control course. The reason for this dual selection was to observe how the freshmen react to graphical programming, likely unknown to them, compared to the traditional way of programming they are more likely to be familiar with.

Various Software Engineering Course Projects: The Software Engineering and Robotics Lab at Florida Gulf Coast University has been in operation for a number of years. It has recently evolved into a sophisticated undergraduate laboratory, with approximately two dozen continuously active student projects, focusing on robotics and embedded systems. Projects selected for presentations in this study include the following:

- NAO humanoid robot project with Kinect vision
- Internet based Hydra gaming platform, and
- 2D game developed using the Unity3D game engine.

All three projects have a significant graphical component included for comparison with LabVIEW.

Data Acquisition and Control Projects: These projects were completely programmed in LabVIEW and involved developing code in block diagrams and corresponding graphical user interfaces. Three projects were presented, including:

- programming a rain gauge connected to a PC via a serial port
- a number of measuring devices and actuators connected to the PC via a USB channel, and
- wireless sensor network remotely accessible and remotely programmable in LabVIEW via a gateway.

It is important to notice the selection of three different hardware interfaces: serial, USB and wireless. This choice was meant to show to the freshmen the spectrum of practical possibilities of making connections to the real world, from the simplest (serial) to the most sophisticated (wireless).
3.0 Detailed Projects Description

This section provides some details on the level of sophistication of the projects as presented to the freshmen.

3.1 Software Engineering Course Projects

The NAO humanoid robot project with Kinect vision focuses on developing a natural uses interface (NIU). The Microsoft Kinect is a sensor based around a color camera, an infrared depth sensor, and a multi-mic array. It allows controlling computing devices via motion and voice instead of mouse and keyboard or joystick, making the interaction much more natural.

The goal of the original project was to be able to control all of the limbs of the robot via motion instead of having to use more complicated control systems. In short, instead of having to manipulate a controller of some sort one could simply move an arm into the position one would like the NAO robot to move its arm, capture this with Kinect, and the software would handle moving the NAO into the correct position, see Figure 1.

The goals of the demonstration to the freshmen class were: (1) to make students aware of the NUI technology, and (2) outline the basic features of programming technologies for graphical applications.

![Figure 1: Kinect vision device (lower left corner) captures the arm movement to control the NAO robot arms (upper right corner).](image)

The Hydra network game project relies on using a proprietary Parallax board connected to the Internet, enabling multiple players to pursue a simple multi-player game. A simple interface is shown in Figure 2. It consists of two parts: a User Interface and a Developer Interface. User Interface (shown on the left hand side) corresponds to playing a multi-player network game and can be more or less sophisticated, depending on the type of the assignment. This is what the student sees when the game software uploaded to the station is run. Developer Interface (shown on the right-hand side) allows uploading the compiled code of the game, zipped image files for the game graphics, and the user manual.
The goal of the original project was the design of the game involving multiple components corresponding to full system as it might occur in a real life: (1) multicore ARM processor, (2) USB bus, (3) bare machine without OS, (4) TinyBasic for programming, (5) vendor specific IDE, (6) HTTP protocol, (7) ASP.NET web technology, and (8) GUI application.

The goals of the demonstration to the freshmen class were: (1) illustrate the diversity of issues facing software developers in a real application, and (2) outline a different programming technology for graphical applications.

The Unity gaming platform is a game development environment incorporating a rendering engine fully integrated with a set of tools to create interactive 3D and 2D content, intended for independent developers. 2D and 3D simulations or visualizations can be built with Unity engine for web sites, desktop platforms, consoles, and mobile devices, see Figure 3.

The objective of the original project was to use a significant application, such as a network game, to verify how a two-semester sequence may reflect on the quality of the final product. The students worked on a team to design and develop the software requirements and related documents in addition to building the software.

The objective of the demonstration to the freshmen class was twofold: (1) illustrate the process of software development comprising multiple stages over two semesters, and (2) outline typical game programming technology with the use of professional tools.
3.2 Detailed Project Descriptions

The data acquisition and control projects were all developed in LabVIEW and focused on a completely different kind of applications, interacting not only with a user, operator or player, but with mechanical or electronic devices, such as actuators and sensors. Students were introduced to the concept of a cyberphysical system, see Figure 4, in which the computer (now understood as a Controller) does not only interact with the User and the Network, but is also aware of the external environment (commonly called a Plant), with which it exchanges data by acquiring results from measuring devices and sending respective signals to the control devices.

![Figure 4: High-level model of a cyberphysical system.](image)

The goals the original projects in the Data Acquisition and Control course were to study the possibilities of programming various interfaces using LabVIEW, and learn how it facilitates device access. This gives the students the broad perspective on connecting to the external world, which is particularly valuable for programming embedded and cyberphysical systems.

The objective of the demonstration to the freshmen was to make them aware of:

- various sophistication levels of interfacing measurement instruments and control devices
- respective standard protocols and implementing the transmissions, and
- existence of a specially designed graphical programming language to handle communication with these devices, with tailored libraries and the need for specific drivers.

The presentations focused on explaining numerous ways of interacting with measurement and control devices, and respective protocols. An interface as simple as a serial protocol over RS232C involved explanations of a data transfer and a handshake through separate lines (wires). This was compared and contrasted to USB, which is a shared bus that transmits both data and control signals by multiplexing them over a single wire.
Then, a wireless Zigbee protocol for National Instruments wireless sensor network was explained as yet another principle, essentially broadcasting information over a different medium. As an example, a single illustration of a LabVIEW program, called Virtual Instrument (VI), is shown in Figure 5 for communication over the serial port. Only the essential VI component is presented, with other necessary logic left out due to the complexity of the diagram.

3.3 Results

Two FAST sessions were given during the semester. The first was given towards the beginning of the semester and the second was given close to the end just before final exam week. In the first session the senior students were from our, Data Acquisition and Control course. They mostly presented LabVIEW based projects. Unfortunately LabVIEW was not installed on the classroom computer and so a live demonstration was not possible. Four teams presented their projects. The second FAST session consisted of two teams and they came from our capstone course called Senior Software Engineering Project. One of the team members in this session started with a small magic show to get the students interested.

The students in the freshmen course was asked to submit a reflection paper for each session. The reflection paper assignments had a weight of 5% of the final grade for both. The students received a full grade if they submit the paper regardless of what the paper says. It was made very clear that this assignment was for them to give us their honest opinion regardless of what that may be and that we needed it to determine if we should continue to have such session in the future. They were reminded that even a paper with the words “no comment” will receive a full grade. The survey was left open ended so as to not influence the student’s responses. A survey with specific question would have given us concrete data however it would also have led the students to simply answer the questions and not tell us about other thoughts they may have. Experience has shown us that the questions for “any additional comments” never get used.

The data we present next in Table 1 was generated by reading each paper and looking for phrases that specifically say they found the session useful to them. Some phrases and our
classification are presented later in the paper. The values -1, 0, and 1 was given to each of two categories, the presentation of the teams and the question and answer session they gave after the presentations. A value of 1 was given if the student found use in the session, a value of 0 was given if the student was indifferent and a value of -1 was given if it was felt the session may have actually harmed the student. For example this paper was classified as "-1" for the presentation part: “The first half of the experience was confusing, slow and not interesting but, after the question round started, I got really into the conversation with the senior student.” Note an average value of 0 indicates the effort had no effect but did not cause damage. There were 39 students in the class. The following data was collected.

Table 1: The data from the two sets of reflection papers evaluated by the authors.

<table>
<thead>
<tr>
<th></th>
<th>Session 1</th>
<th></th>
<th>Session 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Presentation</td>
<td>Q&amp;A</td>
<td>Presentation</td>
<td>Q&amp;A</td>
</tr>
<tr>
<td>no input</td>
<td>0.4286</td>
<td>0.9333</td>
<td>0.7241</td>
<td>0.5926</td>
</tr>
<tr>
<td>-1</td>
<td>4</td>
<td>8</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>2</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td>29</td>
<td>23</td>
<td>16</td>
</tr>
</tbody>
</table>

Observation:

1. The value of 0.4286 indicates the results could have been better but was still a worthwhile effort. Recall the value could go negative. Note there 23 out of 39 students indicated the presentation of this session help motivate them to stay in engineering.

2. The value of 0.9333 indicates the question and answer part of the session was the most useful to them in the first session. However in the second session the presentations were more useful than the question and answer part. This indicates the presentations are different and the students can receive value in them even in multiple sessions however the question and answer part tend to be limited in how many sessions are beneficial. The size and diversity of the senior student also influence the outcomes.

3. Eight (8) students in the first session indicated they got confused by the presentation. Seven of them did not say it cause them to get unmotivated but rather they were simply confused by the presentations. They were labeled as -1 because of the potential it may have to harm the student’s motivation. Care must be taken as a poorly executed session can potentially cause more damage than good. Only one student specifically said the presentation caused intimidation. That student was not an engineering major.

4. The increase in value in the presentation part of session 1 compared to session 2 (0.4286 for session 1 and .7241 for session 2) may have been contributed to the fact that the students in the second session had more experience since they were closer to graduating. This shows that the selection of senior student teams can have an influence on the effectiveness of the program. Care must be taken to ensure the senior students have experience and can give a clear presentation of a working project. The student comments also reflect this.

5. The drop in value between the first and second session (.9333 for session 1 and .5926 for session 2) for the question and answer part may be due to the fact that all of their questions
were already answered in the first session. The number of senior students in the second session was also half.

The following are some excerpts from the student’s reflection papers. These are not a set of excerpt carefully selected to promote our work but rather these excerpt were extracted from 25 of the 52 reflection papers we received. Each excerpt came from a different reflection paper.

- “The visit from the senior students in the software engineering program was very educational. It answered a lot of unknown aspects about the program that didn’t know. I really like having them come in and tell us what to expect. I am also very excited to get to the more advance classes of software engineering. I especially liked all the helpful tips that the students gave.”
- “Before this visit, I was skeptical whether or not I should try to get an internship. But after the visit, I now know that I should try to get one and I will make that a goal of mine.”
- “The senior students got me motivated. If there is one thing that I personally thought was the best attribute from the presentation was the Q/A at the second half of the class. It really got me excited to see what I would be doing and accomplishing in a couple of years of hard work and I just got very motivated to do well in my studies.”
- “Thanks to them, I am now more motivated than ever to remain in the engineering field.”
- “It left me wanting to learn more about this certain field. Overall I was moved by it and would like to remain in the engineering field.”
- “It shows us what’s to come in the future of the program. It lets us see where we ourselves might be and give us time to plan for what’s ahead.”
- “I got to see what I could be doing if I continue down this major. I also saw how what some of what we learned applies to the ‘big picture’. It also opened ideas and opportunity for my future and what major track I take.”
- “Seeing that they were just normal students like me put me at ease and made me feel like I belong in this program. The seniors were very passionate about the FGCU Software Engineering program and this made me extremely excited to be a part of it.”
- “As a freshman, I don’t have classes with upperclassmen that have learned from experience to tell me what’s ahead but the seniors today did a good job.”
- “The round of questions at the end reinforced my desire to stay in the software engineering program.”
- “I was able to learn a lot about how important internships are, and how crucial it is to get your name out there.”
- “I learned so many great things about software engineering it maybe reconsider my major in CIS and change it to software engineering.”
- “The students encouraged me to remain in the engineering field because I like those subjects that they were talking about, and I would like to work for IBM, amazon, google, or any other company.”
• “We can learn the knowledge from class, but we cannot learn experiences. After the visit, I feel I am in this program for something that I can do in reality in the future. Overall I am looking forward to more student visits.”

• “I’ve been told about internships and how important they are, but having a senior from FGCU having already been hired at IBM full time gave me a little more inspiration to get a head-start on internships.”

• “Hearing all of this made me really excited to what there is to come for the rest of my time in college. Being able to hear from students taking the higher-level courses really gave me a view on what was to come in the future.”

• “As a computer information systems major the student visitors definitely peaked my interest on software engineering.”

• “It just shows how quickly the field is evolving and that as a software engineer, my worth will be in constantly evolving and learning new things.”

• “Yes they encouraged me to stay in the software engineering field.”

• “As a mathematics major, it definitely caught my attention since I did not know that there were that many directions that you could go with computers in general. In addition having a background in mathematics does not seem necessary but very helpful and beneficial to the different areas of computer science in general.”

• “It truly got me excited and even got me motivated to continue in the software engineering major and continue my studies.”

• “The visit by seniors could be nothing but beneficial to future software engineers. The students get a first-hand look at the work they are doing, who they are working with and what opportunities they have when they leave here – all three of those points should be taken by students and internalized in such a way that they really can see if this is the path they will ultimately follow.”

• “Overall I was insightful and makes me want to stay in engineering.”

• “I have found I feel much more confident about my decisions when I talk to those that have already made the decision before me. So, for me, this is beneficial. I feel more interaction with seniors is needed to further increase the confidence of those just starting to make a jump into an engineering field.”

• “It is good to know that it's not too late for me to start in programming.”

Data about the visits were collected from the freshmen as well as from the seniors. There were only a few senior teams demonstrating their projects, but a few comments they provided are also important regarding their perception of the class visit.

• “The freshman seemed to respond more to the other projects we showed them such as the NAO robot more than the LabVIEW project. […] They asked several questions about how the NAO robot interacted with the Kinect. After the presentations we stayed and let the freshmen ask general questions about the program. This received the greatest response from the students with many of them asking questions about the program, and how much we enjoyed the program […] In the future I think that showing a broader look at what we have done across several classes would be more interesting and useful to these students.”
• “During the visit to the freshmen class, the students showed interest in embedded systems and microcontrollers such as Raspberry Pi and Arduino. Some of the more inquisitive students asked about future job prospects in the Southwest Florida area and in Florida. Some other students asked about mathematics recommended or required for the degree. A lot of students seemed curious about which programming language is used in the Software Engineering program the most. Another student asked about the G programming language that LabVIEW uses. The class seemed intrigued on a graphical program language.”

4.0 Conclusions

The primary objective of this work was to verify the usefulness of the new didactic method, FAST Learning (Follow Accomplishments of Senior Teams), to attract and retain students in Software Engineering. The method relies on presenting results of senior projects to students in lower level courses, thus, generating or increasing interest in the program among the targeted population. The ultimate question one has to ask is: Did FAST Learning prove to work as assumed and expected?

While the initial results are promising, and the data collected from the freshmen are mostly positive and encouraging, it is too early to tell whether the method guarantees success. Based on students’ comments, one can tell that their level of interest and excitement about the program has increased, so did the level of confidence in pursuing the software engineering major. It also seems that in addition to the main objective, several lower-level goals, such as making students aware of the distinction between traditional textual programming and graphical programming worked well as an attracting factor, so did the diversification of data acquisition projects across the standard interfaces, from the simplest serial port to the much more sophisticated wireless protocol.

However, tracking college careers of the freshmen involved in the study would give more convincing evidence about the project results. This was not possible, though, within a single semester this project was conducted. A broader study is needed to derive more meaningful conclusions, how well the method might work in practice, and such study is currently underway as an extension project.

One additional point to make about this work is that the traditional educational systems, almost exclusively use a bottom-up approach in teaching, that is, from general knowledge, first, to details how to use this general knowledge next, which is also true in software engineering9–10. The FAST Learning method definitely represents something opposite, that is, a top-down approach. Students are presented with solutions first, and learn by generalization, that is, proceed from detail to abstraction.

ACKNOWLEDGMENT

The following students in the Software Engineering program at Florida Gulf Coast University are gratefully acknowledged for their contributions to the projects: S. Curry, M. Evans, S. Gafford, A. Hughes, E. Hutchinson, S. Pais and R. Sinjari. Thanks are due to Dr. Anna Koufakou for using information from one of the projects in her software engineering course.
REFERENCES


