Team Teaching:
Blending the Power of the Socratic Method with Traditional Pedagogy

Tom Goulding and **Durga Suresh**

Department of Computer Science
Wentworth Institute of Technology
gouldingt@wit.edu
sureshd@wit.edu

Abstract

This case study discusses the introduction of the Socratic Method into a traditional computer science classroom. We begin by contrasting the instructional style found within traditional pedagogy with that of a Socratic practitioner. We subsequently present a case study in which juniors being educated in a traditional manner were called upon by a visiting Socratic practitioner to develop the German WWII ENIGMA encryption system in assembly language in five weeks. We hope to confirm in this paper that Socratic Methods can virtually assure success once the power of independent learning and self-reliance is unleashed to create a "failure is not an option" culture in the classroom.

Keywords: Software engineering, game development, rapid development methodologies, computer architecture, encryption systems, independent study.

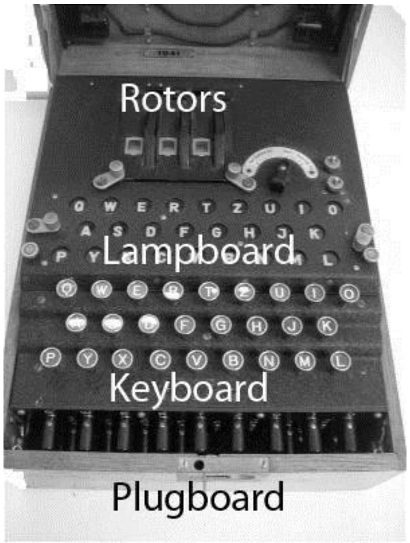
Categories and Subject Descriptors
D2.0 Software Engineering, D3.2 C#

1.0 Introduction

Many educational institutions are now incorporating "real world" projects into the classroom. However, the use of non-traditional educational methods still has not captured the imagination of academics in spite of an illustrious history of success in both ancient and recent times [7,8, 9, 10]. Nevertheless, we hope, herein, to give further encouragement to computer science professionals to consider many of the innovative pedagogical techniques that permeate the research literature. In particular, we hope once again to demonstrate that student success developing complex "real world" software projects, is a likely outcome if a Socratic like pedagogy is adopted in the classroom.

We have previously shown [2, 6] that encryption systems and game projects provide an instructor with the opportunity to challenge students to reach a very high level of software programming achievement while at the same time providing the instructor an opportunity to explore the power found in Socratic pedagogy.

FIGURE 1
GERMAN WWII ENIGMA



Requiring students to recreate the ENIGMA functionality in assembly language also assures that our students are also being called upon to reach toward the highest level of performance in software engineering. One simply cannot develop a product of this complexity in assembly language without being a master at problem decomposition, hardware architecture, modular software design and of course debugging skills.

Launching the ENIGMA project with students inculcated only in traditional lecture-examination pedagogy provides a unique opportunity to demonstrate the power of the Socratic method. The Socratic Method, while of ancient origin, is still remarkable in its ability to unleash the power of independent learning, student initiative and teamwork, even amongst students who have by and large been educated under traditional lecture-examination methods.

Finally, We also believe that a departure from traditionalist lecture oriented pedagogy can have a significant impact on speed at which complex software can be developed. Thus, in this paper we discuss a case study where we challenged the students to develop a completely functional ENIGMA in 5 weeks.

Before we undertake to explain how this is accomplished we begin by highlighting the key aspects and benefits of a Socratic type method..

2.0 The Socratic Practitioner

Socratic pedagogy continues to be largely absent in the traditional campus classroom. Nevertheless, online education, which is inherently Socratic in its approach, has become a dominant force on the educational landscape. We believe the success of online educational models will soon revive interest in Socratic methods in the traditional campus classroom. So what characterizes a Socratic practitioner?

The Socratic instructor believes there is a body of truth external to student experience. In the Socratic tradition, truth is discovered through discourse, otherwise known as the elenetic process. Socrates believed that through asking a series of directed questions his student could arrive at the truth. Thus, the main mission of the Socratic instructor is not to convey the truth, but to help a student find the truth.

The Socratic instructor plays the role of coach, mentor and guide much like a product development manager in a technology company. The primary gift of both an engineering manager and a Socratic instructor is not that of subject matter expert. It is exceptional organizational, motivational, and communication skills that matter to both. A Socratic instructor is skilled at problem definition and decomposition, guiding the search for applicable knowledge, making keen observations, and finally, asking questions that illuminate the path to truth without revealing the truth.

This is perhaps the reason the Socratic instructor remains a rare species in the campus classroom. It is not easy to become an apparent bystander to student learning while at the same time an astute guide. He or she must have confidence in those he or she is guiding. A strong leader defines what must be done and when it must be completed. The road map and resources are provided by the Socratic leader, but those following the Socratic practitioner must scale the mountain themselves.

2.0 Selecting a Motivational Project

We note, however, that an aspiring Socratic practitioner will find his leadership challenge much easier if his or her students are challenged with a project that is inherently interesting, obviously challenging, relevant to their educational and professional goals. The WWII German Encryption System project is such a project.

Student projects conducted in a Socratic classroom build passion, create a strong work ethic, cultivate an ability to rapidly assimilate new knowledge and finally increase self-confidence. We have seen that cultivating these professional attributes requires an educational project that largely imitates the realism and relevance found in industry. We believe the Socratic method imitates corporate environments far better that traditionalist classroom methods and therefore students are more likely to perform with the motivation and energy typically found in engineering organizations. The pace is fast; the learning primarily done independently and the mastery and application of new knowledge rapid. ENIGMA is a project that creates these type of learning opportunities.

3.0 Launching the Assembly Language Class

The assembly language class where this trial was being carried out consisted of upper level computer science majors studying assembly language using the Intel 32 bit op code set. The traditional approach to the class involved detailed faculty lectures on data structures, formatting and encoding, opcode mastery and instruction set usage. Class dialogue and examination questions focused on validating student understanding of the wide range of important hardware and software concepts that must be mastered when programming at the lowest level in a computer system. A complex project had not been previously considered for this type of course at Wentworth Institute of Technology.

The plan which the authors adopted involved the Socratic instructor (Professor Goulding) joining the class as a guest lecturer eight weeks after the course began. At this point most of the students would have had reasonable familiarity with the Intel opcode instruction set.

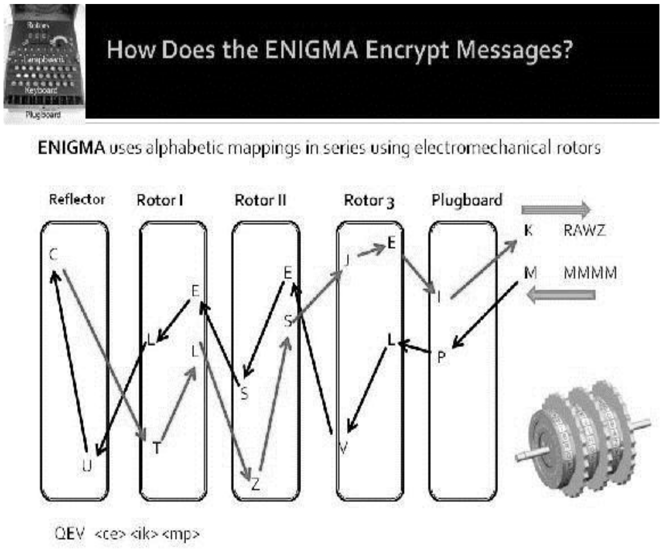
The Socratic instructor assiduously avoided discussing the Intel instruction set. However, in the Socratic style guided the students into an understanding of the ENIGMA functionality as well as a software architecture for the project.

4.0. How the ENIGMA works?

The German WWII ENIGMA system encrypts alphabetic messages in a manner that is reminiscent of the ancient, well known and rather simple Caesar cipher. The ENIGMA, however, is a complex electro-mechanical device (Figure 1) that maps one alphabetic character to another. The system takes a character through a series of fixed alphabetic transformations. Using 5 subsystems, namely, a plug board, three rotors and a reflector, the ENIGMA forces a character through 24 transformations before the final encrypted output character is achieved.

As illustrated in Figure 2, the character M will be encrypted to the output character K after 24 different transformations (not all shown). If a series of M's are typed each M will encrypt to a different character.

Figure 2



How Does the ENIGMA Encrypt Messages?

ENIGMA uses alphabetic mappings in series using electromechanical rotors

MMMMM becomes KRAWZ. In the different configuration (Figure 4) the ENIGMA would encrypt MMMMM to YZEBLP. This was quite a mystery to the code breakers of Bletchley Park. [1] Eventually, they would discover that the rotors turn after each character is typed. This ensures that the internal alphabetic mappings change after each character is entered on the keyboard.

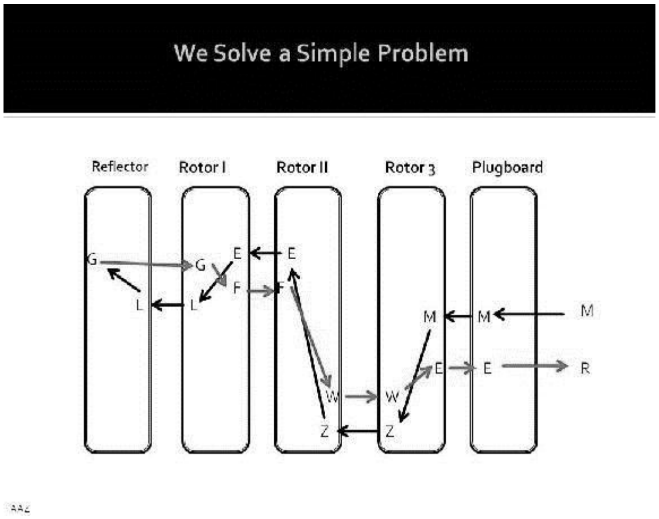
5.0 Guiding the Project Using the GM-Method

The pedagogical method, used to guide Professor Suresh's students is a Socratic variant that uses best practices from both educational and engineering development settings.

Our variation of the Socratic method has been under development for ten years. The method called the GM-Method has been discussed extensively in previous papers. [2 - 6] Therefore, we will discuss the method in a very abbreviated manner and only in so far as it informs this case study.

However, we do note that the key to success for a rapid development cycle discuss herein was to help the students understand the importance of problem simplification.

Figure 3
A simple ENIGMA Configuration



We Solve a Simple Problem

Figure 3, for example, illustrates where the experience of the Socratic instructor becomes essential to student success. With appropriate simplifications rotor rotation can be eliminated and the ENIGMA can then be reduced to a simple table driven architecture involving only 9 transformations. This basic architecture then becomes the foundation upon which the complexity and eccentricity of the ENIGMA system can be built.

Without the Socratic instructor providing this simplified beginning point it is doubtful any pedagogical method could lead students to success in a 5 week interval.

Once students succeed with this very simple ENIGMA transformations illustrated in Figure 3, they begin to believe they can achieve what appeared to be an impossible task when their five week journey began. Much like constructing a house, this simple foundation is laid first and then the students soon add the walls, roof, windows and architectural amenities. Five basic laboratory exercises build upon this simple architecture and provide the road map that leads to the summit, namely, a completely functional ENIGMA including in famous design flaw the double step.

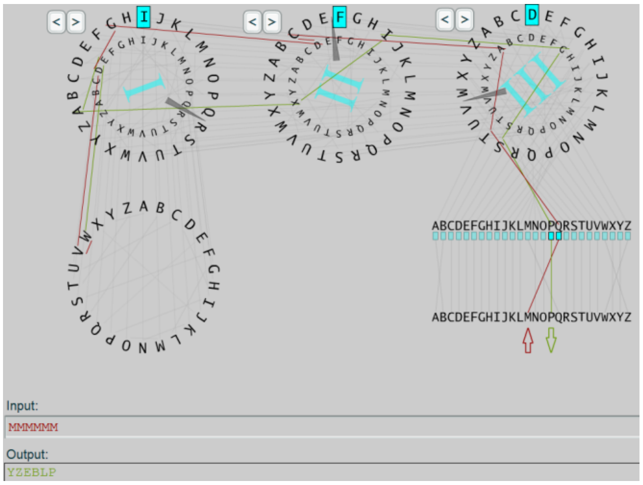
6. 0 The Outcome

One might ask whether the student encryption system depicted in Figure 5 matches identically the workings of the German ENIGMA. One of the most useful resources available to students to validate their work is a web based ENIGMA demonstration (Figure 4) developed by a German scientist, Dr. Frank Spiess [14]. Without this demonstration of rotor behavior, students would either have to secure a historical copy of the ENIGMA system, which is a virtual Impossibility, or they would have to uncover the subtleties of ENIGMA rotor interaction by reading specifications and research reports. Either, of course, would be too time consuming and probably beyond the reach of even the most energized students working against a five week project development cycle.

Comparing the result in Figure 4 with that of Figure 6 shows that Frank Fodera's student implementation reproduces exactly the encrypted output of Dr. Speiss' web based demonstration of the ENIGMA under identical initialization conditions, (which are not evident by looking at the images.) The output YEZBLP is the result of both simulators.

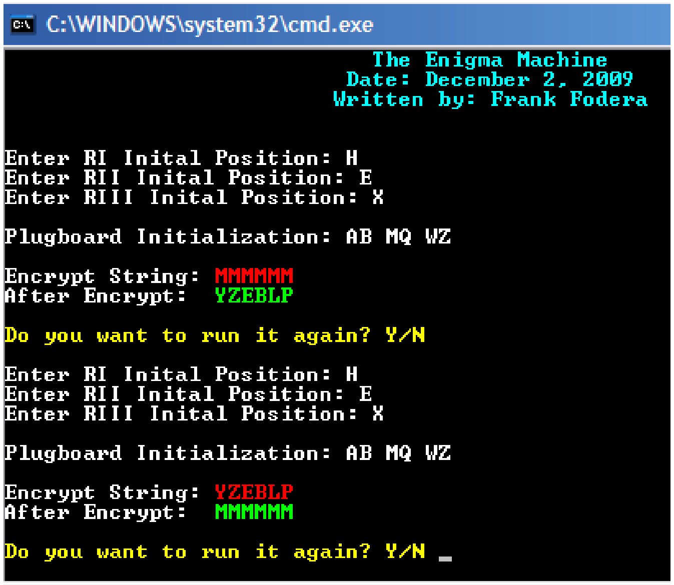
Student success confirms once again that the critical success factor is adopting a pedagogical method that unleashes student initiative and passion. While initially believing the project to be impossible, Frank Fodera caught the vision and according to his own words 'stopped going to the cafeteria and out with this friends until he completed the ENIGMA. The impossible became the possible and he made it happen. Approximately, 50% of the class had the same level of success and completeness as Frank.

Figure 4
Dr. Speiss'
ENIGMA Demonstration



Input:
MMMMMM
Output:
YZEBLP

Figure 5
Student Frank Fodera
WW II ENIGMA



C:\WINDOWS\system32\cmd.exe

```
                                       The Enigma Machine
                                     Date: December 2, 2009
                                   Written by: Frank Fodera

Enter RI Inital Position: H
Enter RII Inital Position: E
Enter RIII Inital Position: X

Plugboard Initialization: AB MQ WZ

Encrypt String: MMMMMM
After Encrypt:  YZEBLP

Do you want to run it again? Y/N

Enter RI Inital Position: H
Enter RII Inital Position: E
Enter RIII Inital Position: X

Plugboard Initialization: AB MQ WZ

Encrypt String: YZEBLP
After Encrypt:  MMMMMM

Do you want to run it again? Y/N _
```

Developing a fully functional WWII Encryption system is quite a professional feat for any undergraduate. It builds self confidence, self esteem and awareness that "impossible" problems can be solved if they are simplified and attacked in an incremental manner. Hard problems are usually solved by working on a series of simple problems. It is an important lesson for students to learn.

We believe the development of a significant software project in is assembly language is especially important. However, finding interesting "real world" projects that can be achieved in assembly language under very tight time constraints is not easy. ENIGMA seems to fill the bill. Others could be creating the call processing software for a circuit switch or the networking and data link layers for packet switches.

5.0 CONCLUSION

We hope to have demonstrated that students with little to no complex project experience were able to quickly apply their basic technical knowledge to solving a very complex systems problem.

Having a very challenging "real world" project is energizing and creates the work ethic and motivation necessary to lead to student success.

The Socratic method captures the essence of the development culture found in industry and therefore becomes one of the most desirable pedagogical methods to use in the engineering and computer sciences.

References

[1] Bletchley Park National Codes Center, http://www.bletchleypark.org.uk/

[2] Goulding T., (2008) "Complex Game Development throughout the College Curriculum" *Association for Computing Machinery,* SIGCSE Bulletin (December 2008), Volume 39, 68-71.

[3] Goulding, T. Using Complex Game Development to accelerate learning in the CS1 and CS2 Programming Courses. *IMSCI 08, 2th International Multi-Conference on Society, Cybernetics and Informatics Proceedings, June 2008, 163-168*.

[4] Goulding, T. & DiTrolio, R. Complex Game Development by Freshman Computer Science Majors. *Inroads,* Association for Computing Machinery, SIGCSE Bulletin, *39,* December 2007, 92-97.

[5] Goulding, T. & DiTrolio, R. Complex Game Development: A Case Study in Rapid Software Development By Novice Programmers. IASTED: SEA07 Conference Proceedings. MIT Cambridge, MA, November 2007, 295-304

[6] Goulding, T. & DiTrolio, R. Introducing Realistic Constraints into Student Project Teams. *Inroads,* Association for Computing Machinery, SIGCSE Bulletin, *37,* December 2005, 54-55

[7] Green, P. The Potential of Gaming on K-12 Education. *MultiMedia & Internet2Schools, 13*(2), (May 2006) Retrieved March 25, 2008 from **http://goliath.ecnext.com/coms2/gi_0199-5573907/The-potential-of-gaming-on.html#abstract**

[8] Gow, L. & Kember, D. Does higher education promote independent learning?, *Higher Education,* (19), (1990). 307-322.

[9] Gupta, M. Enhancing student performance through cooperative learning in physical sciences. *Assessment & Evaluation in Higher Education,* 29 (2004), 63-73. Retrieved on March 5, 2007 from the Academic Search Premier database

[10] Holubec, E., How do you get there from here? Getting started with cooperative learning. *Contemporary Education*, Vol. 63, (Spring,1992), 181-184

[11] Jones, F. The Moore method. *America Mathematical Monthly, 84* (1977), 273-277

[12] Long, C., Educators Got Game, NEA Today, 28(2), (2007) 42-43.

[13] Patt, Y & Patel, S Introduction to Computing Architecture 2nd ed. McGraw Hill, 2008

[14] Spiess, F., Enigma Flash Retrieved July 29, 2009 from **http://enigmaco.de/_fs/index-enigma.html**

[15] Tucci, L., College Students Continue to Shun Computer Science, SearchCIO.com June 8, 2005, Retrieved March 25, 2008 from **http://searchcio.techtarget.com/news/article/0,289142,sid182_gci1096260,00.html**

[16] Trotter, A. Despite Allure, Using Digital Games For learning, *Education Week*, *25* (10), (2005), 1 – 19.