ABET Cybersecurity Continual Course Improvements for Secure Software Development

Suzanna E. Schmeelk, Denise M. Dragos, and Joan E. DeBello
Dept. of Computer Science, Mathematics, and Science
Collins College of Professional Studies (CCPS), St. John's University
New York City, United States of America
{schmeels, dragosd, debelloj}@stjohns.edu

***Abstract*—This is an innovative practice full paper. The need to develop software securely cannot be over-emphasized. The changing legal and regulatory international and local landscape for software requirements is astounding. For example, the European Union's General Data Protection Regulation (GDPR), the United States' Health Insurance Portability and Accountability Act (HIPAA), the Chinese Cybersecurity laws, and the credit card industry's Payment Card Industry Data Security Standard (PCI-DSS) are all upholding higher standards for system development and deployment. Such legal and regulatory changes of necessity require modifications and updating in software development methods that must be incorporated into cybersecurity software development courses to properly prepare students for successfully working in the field. To address these and other changes within the computing field, the Accreditation Board for Engineering (ABET) recently proposed preliminary cybersecurity accreditation criteria for which fewer than 20 universities have both applied and become ABET Cybersecurity accredited. The accreditation requires maintaining continuous course improvement in the core courses including a secure software development course. This research first reports on important topics incorporated into a senior-level secure software development for cybersecurity majors. Our research then analyses student Institutional Review Board (IRB) approved surveys to learn which course components could benefit from continuous course improvements. We apply machine learning to help build categories for ABET continual improvement. Finally, we share lessons learned and plans for future work.**

***Index Terms*—secure software development, object oriented programming, scripting, cybersecurity, Java language, application development**

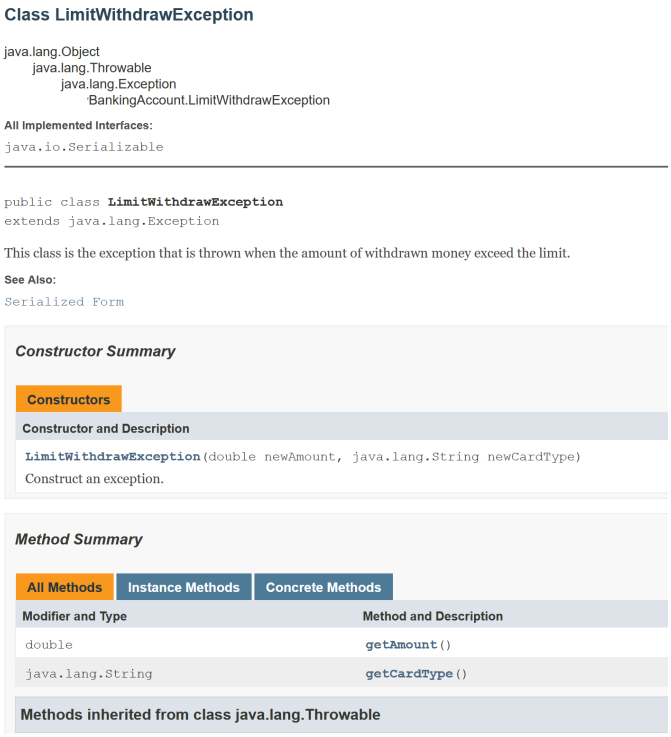
I. Introduction

Cybersecurity has become fundamental to all information networks and systems. The recent ransomware attack on the North East Colonial Pipeline is a stark reminder of the results from the lack of cybersecurity. Among the recent drivers for the importance of cybersecurity are regulations and the protection of factors which go into risk assessments, such as reputation and life protection, among other concerns. To keep pace with the changing cybersecurity industry needs and research, academic institutions have been developing and expanding curricula both to address the real world changes and to prepare students for entry into the international cybersecurity workforce. As the curricula and field have grown, so has the need for accreditation programs that certify the quality of such curricula. One of the foremost accreditation to date is developed by ABET, the Accreditation Board for Engineering and Technology. The contribution of this paper is to begin to fill a literature gap for ABET secure software development curriculum advancement and improvement.

II. ABET Accreditation Literature Review

There are few ABET accreditation research papers. Raj, Anand, Gibson, Kaza, and Phillips [1] led a panel on the discussion of the benefits, challenges, and costs of cybersecurity program accreditation. Maymí [2], an industry professional, gave a keynote discussing topics he hopes are being taught and learned in academic settings. Parrish, Impagliazzo, Raj, Santos, Asghar, Jøsang, Pereira, and Stavrou's [3] research discussed a futuristic and industry-preparatory need for cybersecurity education and propose a consistent terminological framework around a vision for education. Coello and Huggins [4] presented a methodological proposal to carry out the measurement of student outcomes recently published by the Engineering Accreditation Commission of ABET.

Helps, Lunt, and Anthony [5] described their experience at Brigham Young University of preparing for an ABET IT accreditation visit. They discussed issues of continuous improvement, assessment, accreditation documentation, and methods for collecting and presenting accreditation data. The authors discussed the selection of program objectives, which were related to their institutional objectives, and their links between course outcomes and program outcomes and objectives. Greenlaw, Phillips, and Parrish [6] discussed the increase in interest in cybersecurity and related areas, which they cited as information assurance and cyber operations, and suggested/questioned adding an ABET cybersecurity accreditation. They then provided the foundational research for two questions: "What constitutes a sufficient case for rolling out new accreditation criteria?" and "Should new criteria for cybersecurity be developed?" To answer the questions, they started by developing a checklist of conditions based in part on the National Security Agency (NSA) criteria for a Center of Academic Excellence (CAE), which they suggested are strong



Class LimitWithdrawException

java.lang.Object
  java.lang.Throwable
    java.lang.Exception
      BankingAccount.LimitWithdrawException

All Implemented Interfaces:
`java.io.Serializable`

`public class LimitWithdrawException`
`extends java.lang.Exception`

This class is the exception that is thrown when the amount of withdrawn money exceed the limit.

See Also:
`Serialized Form`

Constructor Summary

Constructors

Constructor and Description
`LimitWithdrawException(double newAmount, java.lang.String newCardType)` Construct an exception.

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method and Description
`double`	`getAmount()`
`java.lang.String`	`getCardType()`

Methods inherited from class java.lang.Throwable

Fig. 1. Curriculum Example: An example API Documentation Created by Students

indicators for the successful addition of new ABET criteria. These conditions are: maturity, expertise, champions, importance, timeliness, demand, well-defined, qualified evaluations, interest level, fundamental relevance, and separate discipline.

Byrd's [7] research goal was to help clarify what math topics and courses would be most helpful in preparing students for a program of study in cybersecurity and eventually for success in that professional environment. Byrd reviewed six primary sources for mathematical needs, specifically: Certified Information System Security Professional (CISSP) Common Body of Knowledge (CBK) domains, ABET draft cybersecurity accreditation criteria (CAC), National Security Agency (NSA)/Department of Homeland Security (DHS) Centers of Academic Excellence, Association of Computing Machinery (ACM) standard computer science curriculum, and National Initiative for Cybersecurity Education (NICE). Byrd found mathematics relevant to each domain.

III. Secure Software Development Curriculum

The Secure Software Development curriculum consists of best practices for both secure scripting as well as secure application development. Topics in the course currently include building custom exceptions, number overflows, input validation, output encoding, auditing, generics; database queries employing prepared statements; building custom Application Programming Interface (API); static analysis; and specific digital forensics scripting principles. The purpose of the course is to introduce students to secure application development principles as well as to secure scripting principles.

In the course, students spend approximately ten weeks on building an object-oriented application and learning corresponding security concerns. They then spend approximately five weeks on secure scripting with PowerShell and Python. Students prepare and present on security concerns with relevant scripting languages and report on security breaches, including examining concerns listed on the National Institute of Standards and Technology's (NIST's) National Vulnerability Database (NVD). The final course project is to integrate the labs performed over the first ten weeks of the course (and the feedback received) to build a final report of techniques employed to secure an application.

A. Building a Custom Secure Software Notated API

The students develop security requirements and learn to build their own custom application programming interface (API). One of the requirements for API development is to indicate directly in the API how they made secure software design choices. The students include the documentation into their final report. Figure 1 shows one API class built for the project employing Javadoc and notating secure software design choices implemented.

B. Implementing Input Validation and Output Encoding

In the final report, students are asked to explain how they employed important cybersecurity mitigations such as input validation and output encoding. Figure 2 shows a project by a student who participated in our IRB study in which they employed input validation for secure software best practices as

```
public void setFirstName(final String newFirstName) {
   String s = Normalizer.normalize(newFirstName, Normalizer.Form.NFKC);
   Pattern pattern = Pattern.compile("^[a-zA-Z ]{2,20}$");
   Matcher matcher = pattern.matcher(s);
   if(matcher.find()) {
    this.firstName = newFirstName;
   }else {
    throw new IllegalArgumentException("Invalid first name argument:" + newFirstName);
   }
}
```

Fig. 2. Curriculum Example: Student Project Implementing Input Validation based on CMU-SEI

```
/**@return unencrypted balance.
 * */
public double getBalance() {
  double s;
  try {
   Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
   cipher.init(Cipher.DECRYPT_MODE, this.k, this.IV);
   byte[] m = this.encBalance;
   byte[] b = cipher.doFinal(m);
   s = convertByteArrayToDouble(b);
   return s;
} catch (UnrecoverableKeyException | KeyStoreException |
         NoSuchAlgorithmException | NoSuchPaddingException |
         IllegalBlockSizeException | BadPaddingException |
         InvalidKeyException | InvalidAlgorithmParameterException e) {
   handleEncryptionException();
}
return 0;
}
```

Fig. 3. Curriculum Example: Student Project Implementing Encryption based on CMU-SEI [8], Oracle [9], and OWASP [10]

given by the Carnegie Mellon University Software Engineering Institute (CMU-SEI) [8].

C. Implementing Cryptography

The students learn how to employ best practices for employing cryptography for data-at-rest in Java. Specifically, they employ best practices given by the Open Web Application Security Project (OWASP) [10], CMU-SEI [8], and Oracle [9] researchers for which algorithms to employ along with modes of operations and best practice padding schema. Figure 3 shows sample code from a student's project who participated in our IRB-approved study. Specifically, this project chose to employ the symmetric-cryptography Advanced Encryption Standard (AES) algorithm with the cipher block chaining (CBC) mode of algorithm operation along with the PKCS5 padding schema.

IV. Lessons for ABET Continual Improvement

To develop categories for the ABET (re)accreditation continual course improvement requirements, we surveyed students with an Institutional Review Board (IRB)-approved optional survey. The student responses were clustered using machine learning techniques to identify categories for continuous improvement employing machine learning techniques published by Dragos and Schmeelk [11]. Term frequency-infrequent document frequency (TF-IDF) with a K-Means clustering algorithm was chosen to categorize the cleaned and pre-processed data as it can be useful to standardize and aggregate open-ended and non-structured textual survey feedback.

A. Final Project Benefits

When asked the question, "Q8 - (7) What are/were the benefits to working on the final project? (Please keep responses anonymous.)." there were 21 students who participated in the IRB-approved open-ended survey question in this course. The cluster heterogeneity of their responses is plotted in Figure 4 (a), and the log transformation is presented in Figure 4 (b), showing that heterogeneity is discontinuous from cluster count seven and above; thus, we treat seven as the optimal cluster count.

Table I shows the top five keywords associated with student responses with respect to benefits with working on the secure software development final project.

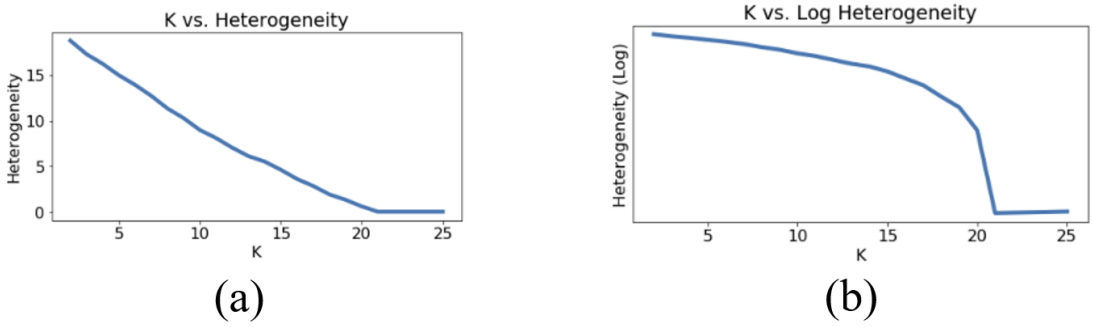


Fig. 4. Benefits to working on the final project term frequency-infrequent document frequency (TF-IDF) with a K-Means clustering analysis algorithm was chosen to categorize cleaned and pre-processed IRB-approved survey data: (a) cluster heterogeneity; (b) log transformation.

#	C	Top 5 Keywords	Surveys	Interpretation
0	5	peer:0.277 team:0.127, set:0.124, base:0.124 work:0.110	3,9, 10, 19 21	Different perspectives on problems approach.
1	1	start:0.327 review:0.327 finish:0.327 perfect:0.327 encompass:0.327	0	Final project encompassed entire course contents.
2	2	teamwork:0.980, learn:0.139 major:0.000 taught:0.000 split:0.000	2,15	Team divided work.
3	4	benefit:0.173 practic:0.159 includ:0.159 appli:0.129 solv:0.113	1,8, 13,14	Applied problem solving.
4	3	faster:0.293 done:0.293 get:0.220 futur:0.194 abl:0.181	16,17, 20	Future work will be solved faster due to experience.
5	3	understand:0.191 concept:0.147 grasp:0.147 didnt:0.147 fulli:0.147	5,6, 18	Team help learn concepts.
6	3	one:0.276 idea:0.199 incorpor:0.167 differ:0.141 anoth:0.140	4,7, 11	Learn from different ideas.

TABLE I
TOP TF-IDF PROJECT BENEFITS KEYWORDS

#	C	Top 5 Keywords	Surveys	Interpretation
0	6	virtual:0.310 experi:0.102 easier:0.099 think:0.092 materi:0.091	3,4,5, 11,17, 21	Availability to coordinate virtually.
1	4	implement:0.188 onlin:0.136 difficult:0.128 cryptographi:0.122 techniqu:0.122	0,6, 8,16	Difficulty with learning cryptography.
2	3	commun:0.621 lack:0.254 commit:0.192 reli:0.192 other:0.149	1,14, 15	Lack of team communications.
3	9	none:0.111 work:0.102 time:0.097 peopl:0.093 set:0.076	2,7,9,10, 12,13, 18,19,20	Coordinating synchronous work times.

TABLE II
TOP TF-IDF PROJECT OBSTACLES KEYWORDS

#	C	Top 5 Keywords	Surveys	Interpretation
0	10	none:0.100 demonstr:0.085 fun:0.085 commun:0.079 onlin:0.077	1,4,6, 7,9,10,12, 14,17,18	Nothing, the community was fun.
1	2	bit:0.575 minim:0.255 fast:0.242 phenomen:0.242 speed:0.242	0,11	Phenomenal class, fast pace.
2	4	group:0.245 project:0.223 portion:0.105 portion:0.105	2,8, 13,15	Portion out final project and final exam.
3	3	great:0.385 class:0.267 complet:0.179 certainli:0.105 cu:0.105	3,5, 16	Great class to become employable.

TABLE III
TOP TF-IDF PROJECT IMPROVEMENTS KEYWORDS

B. Final Project Obstacles

When asked the question, "Q9 - (8) What were the obstacles to working on the final project? (Please keep responses anonymous.)." 22 students in the course responded. The cluster heterogeneity is plotted in Figure 5 (a), and a log transformation is presented in Figure 5 (b), showing that heterogeneity is discontinuous from cluster count four and above; thus, we treat four as the optimal cluster count.

Table II shows the categories of student final project obstacles responses for continual course improvement.

C. Final Project Improvements

Nineteen students in the course responded to the question, "Q10 - (9) What can be improved if the class runs again? (Please keep responses anonymous.)." The cluster heterogeneity is plotted in Figure 6 (a), and a log transformation is presented in Figure 6 (b), showing that heterogeneity is discontinuous from cluster count four and above; thus, we treat four as the optimal cluster count.

Table III shows the categories of student final project improvement responses for continual course improvement.

V. FUTURE WORK

There are many paths for future research. First, we can explore continual improvement assessments in other courses within our Cyber Secure Systems (CSS) program. Second, we can report on the ABET self-study reflections of our program. Third, we can report on our continual improvement

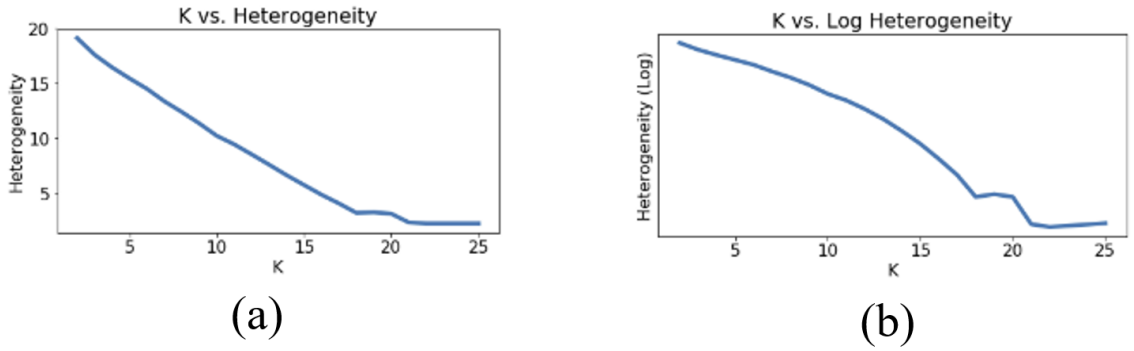


Fig. 5. Obstacles to working on the final project term frequency-infrequent document frequency (TF-IDF) with a K-Means clustering analysis algorithm was chosen to categorize cleaned and pre-processed IRB-approved survey data: (a) cluster heterogeneity; (b) log transformation.

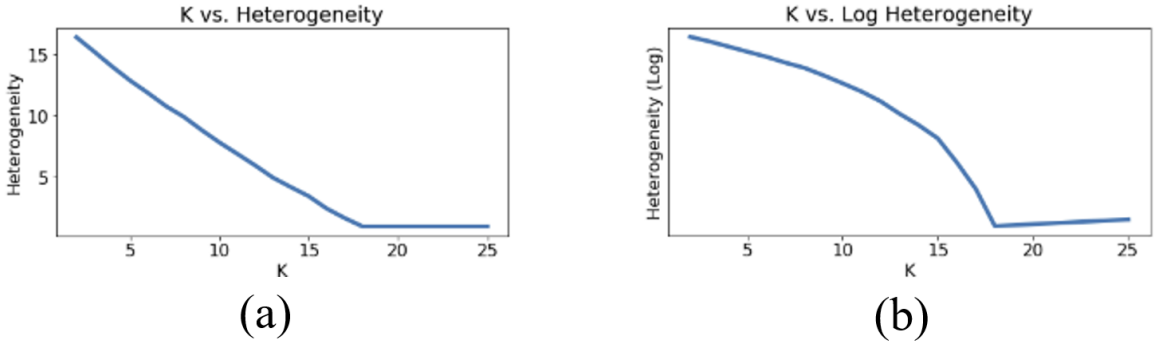


Fig. 6. Course improvement suggestions from term frequency-infrequent document frequency (TF-IDF) with a K-Means clustering analysis algorithm was chosen to categorize cleaned and pre-processed IRB-approved survey data: (a) cluster heterogeneity is plotted; (b) log transformation.

metrics. Metrics included into this category involve changing our course prerequisites, adding additional courses to our curriculum, and/or changing assessment tasks, among others. We can also report on further categories, through machine learning techniques, developed for ABET continual improvement [11]. Lastly, we can study the change in terms and categories from student feedback over time.

VI. Conclusions

The ABET cybersecurity assessment process has been very insightful for different reasons. First, the assessment process motivates faculty to deeply reflect on their courses from the point of view of student outcomes. Ultimately, education should focus on student learning. Faculty participating in the ABET process carefully reflect on student assessment outcomes and continuing curriculum improvements. We have found that creating IRB-approved surveys can help in the collection of continual course improvement metrics as reported by Dragos and Schmeelk [11]. One of the most useful sources of feedback is directly from students which can be integrated in newer iterations of the course.

Second, the assessment process energizes faculty to communicate with each other and the students to plan future topics within each course. For example, in our graduate program, we have begun to coordinate laboratory exercises between courses. This extra planning helps instructors identify student backgrounds prior to course deployment and eliminates laboratory redundancy across classroom experiences.

Third, the assessment process encourages faculty to communicate with students to learn about their difficulties. Again, custom IRB-approved surveys can facilitate the improvement of student needs (e.g. benefits, obstacles, etc.) to make changes for the next course iteration. In addition, faculty leading the ABET process can collect metrics from graduating and alumni students.

Overall, ABET Cybersecurity assessment is a very large time and project commitment requiring institutional support. However, the students' learning is the focus of the work further motivating faculty.

References

[1] R. K. Raj, V. Anand, D. Gibson, S. Kaza, and A. Phillips, "Cybersecurity program accreditation: Benefits and challenges," in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, (New York, NY, USA), p. 173–174, Association for Computing Machinery, 2019.

[2] F. J. Maymí, "Cybersecurity: What i hope someone's teaching," in *Proceedings of the 20th Annual SIG Conference on Information Technology Education*, SIGITE '19, (New York, NY, USA), p. 1–2, Association for Computing Machinery, 2019.

[3] A. Parrish, J. Impagliazzo, R. K. Raj, H. Santos, M. R. Asghar, A. Jøsang, T. Pereira, and E. Stavrou, "Global perspectives on cybersecurity education for 2030: A case for a meta-discipline," in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE 2018 Companion, (New York, NY, USA), p. 36–54, Association for Computing Machinery, 2018.

[4] J. G. Coello and K. Huggins, "The students outcomes abet (1-7) and solo's taxonomy: An approach," in *Proceedings of the 5th International Conference on Frontiers of Educational Technologies*, ICFET 2019, (New York, NY, USA), p. 110–117, Association for Computing Machinery, 2019.

[5] C. R. G. Helps, B. M. Lunt, and D. K. Anthony, "Abet accreditation with it criteria," in *Proceedings of the 6th Conference on Information Technology Education*, SIGITE '05, (New York, NY, USA), p. 353–359, Association for Computing Machinery, 2005.

[6] R. Greenlaw, A. Phillips, and A. Parrish, "Is it time for abet cybersecurity criteria?," *ACM Inroads*, vol. 5, p. 44–48, Sept. 2014.

[7] R. Byrd, "Cybersecurity: 1) what math is necessary and 2) developing ubiquitous cybersecurity in current computing programs," *J. Comput. Sci. Coll.*, vol. 33, p. 53–59, Apr. 2018.

[8] C. M. University, "Secure coding guidelines," in *Software Engineering Institute*, 2021.

[9] Oracle, "Oracle secure coding guidelines for java se.," in *Oracle's Guidelines.*, 2021.

[10] OWASP, "Owasp top 10 risks," in *The Open Web Application Security Project (OWASP)*, 2021.

[11] D. Dragos and S. Schmeelk, "What are they reporting? examining student cybersecurity course surveys through the lens of machine learning," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 961–964, 2020.