https://doi.org/10.48009/3_iis_2018_131-138
Issues in Information Systems
Volume 19, Issue 3, pp. 131-138, 2018

STANDARDS DRIVEN CURRICULUM FOR SECURE SOFTWARE DEVELOPMENT

Andrew Allen, Georgia Southern University, andrewallen@georgiasouthern,edu
Jim Harris, Georgia Southern University, jkharris@georgiasouthern.edu
Vladan Jovanovic, Georgia Southern University, vladan@georgiasouthern.edu

ABSTRACT

This paper describes a proposal to improve curricula to promote secure software development through the application of standards, by providing not only coverage about discipline specific standards, but also using these standards as key course components throughout the curriculum. Being cognizant of the already limited contact hours available during a course, the presentation of content is designed to provide an instructor with the flexibility to select specific modules to incorporate in the course and/or use as additional online self-managed activities that can be assigned to students. While the intent is to be able to use this work in an applied undergraduate program in Computer Science, the developed artifacts can be used in other relevant programs in Computing.

Keywords: Application of Standards, Computer Science Education, Modular, Secure Software

INTRODUCTION

The pervasiveness of computers and connected devices has created a high demand for computing related graduates to develop and maintains these systems. This trend is expected to continue into the foreseeable future. For science and engineering occupations, six of the ten largest employment growths from 2012 to 2016 were computing related occupations (Sargent, 2017). It is, however, more challenging to support and protect these highly connected and complex systems from bad actors, bad code and bad processes, all of which introduce, exacerbate or exploit vulnerabilities. Dr. Konstantinos Karachalios, Managing Director of the Institute of Electrical and Electronics Engineers' Standards Association describes standards as "published documents that establish specifications and procedures designed to maximize the reliability of products, materials and services" as well as "safeguard consumer safety" (Karachalios, 2017). An Enterprise Strategic Group study (Oltsik, McKnight, & Gahm, 2010) reported that over 40 % of surveyed organizations trust their developers to know how to develop secure software. For software development, understanding and applying a standards based approach to the process has been proven to increase security (Sila, 2018) (Amoroso, 2018), It is therefore imperative that standards be taught in the curriculum.

Educators understand the benefits of providing future graduates with an appreciation of the standards in the discipline and the knowledge to apply the standards (Jovanovic, Andres, & Sherlund, 1999) (Jovanovic & Harris, 2016). Almost all Computer Science departments offer security specific courses but most tend to be elective offerings for seniors. The results of a survey of computer Science department heads conducted by Zatko (Zatko, 2016) reports "The results indicate that when security is taught as a separate subject, it's not highly prioritized in the overall curriculum design". Integration of security within existing courses is one way of increasing the prioritization of security and standards within the curriculum. However, with the current demands due to research expectations or teaching loads, some instructors are often constrained by a lack of time to prepare content on standards. Additionally, with the limited number of contact hours per course, instructors have to prioritize topics to be covered which usually results in a lack of time to deliver content on standards.

This work aligns with and supports the "diffusion through curricula" approach described by Yuan et al (Yuan, Yang, Jones, Yu, & Chu, 2016). We present four courses that can seamlessly support standards integration while providing students with opportunities to understand and apply standards as a secured coding practice. We present the curriculum elements that can aid or augment instruction in these four courses. The proposed curriculum elements are designed to help solve the issues of instructional development and delivery times by providing prebuilt modularized content on standards while keeping the instructor in the loop via access to auto graded quiz results of the students.

METHODOLOGY

While the documents for each of the standards that will be addressed in this paper are widely available, understanding and interpreting these lengthy technical documents can be challenging for a first time reader. Recognizing that these technical documents are guidelines and not prescriptive can also be confusing for students. The methodology used to develop the instruments is based on a view that the problems outlined earlier can be addressed with a combination of four components.

1. Instructional Strategy
2. Sequencing
3. Delivery Mechanism
4. Competence assurance

Instructional Strategy: The project team developed, and intends to make available, lecture modules suitable for online delivery, flipped classroom, and experiential process learning educational models. The major themes of each of the initial four standards oriented courses will be supported by one or more modules. In some cases, the modules will cover complete coursework.

Sequencing: The major themes of each standard covered will have material that targets three levels of expectation:

1. Awareness, which will be evaluated as recall;
2. Deduction, which will be evaluated as the inference of the implications of the standards;
3. Application, which will be evaluated as the effective use of standards in a given scenario.

The instructor will be able to assign modules for a single level or multiple levels. The modules will be auto-graded (whenever feasible) and provide an option for the instructor to retrieve the students' scores.

Delivery Mode: The lecture modules developed will be a combination of several delivery modes, inclusive of video, lecture notes, some required/recommended readings (including copies of standards/guidelines, for example selected NIST SP-800 series) and suggested assignments.

Competence Assurance: IEEE defines a *learning object* as any entity, digital or non-digital, that may be used for learning, education or training (IEEE Learning Technology Standards Committee, 2002). For the purposes of this project, a learning object will be limited to digital entities. The learning objects will be developed using the Shareable Content Object Reference Model (SCORM),) (SCORM Explained, 2009), which is considered to be the de-facto standard and used widely, including the DoD. Furthermore the xAPI (Overview of xAPI, 2014) will be investigated as a method of extending the SCORM efforts to a data vault type data warehouse as a Learning Repository Store (LRS) of experience.

The *awareness* and *deduction* online resources will be developed as learning objects. Each learning object will have an objective/description, content, practice quiz and an auto-graded assessment quiz. The *application* online resources will be developed as learning objects that will include instructions, project assignments and examples of graded project assignments. The learning objects will be hosted on the NSF sponsored SEP-CYLE project (STEM-CyLe Team, 2013) at stem-cyle.cis.fiu.edu, a configurable learning and engagement cyber learning environment. Preliminary designs will be evaluated for improvements during 2018 and the effectiveness of the developed modules will be assessed using pre/post-tests of students in 2019.

ORGANIZATION OF MODULES

The initial focus will be on modules within four courses: 1) Computers, Ethics, and Society; 2) System and Software Assurance; 3) Software Testing and Quality Assurance; and 4) Software Engineering a Capstone Project course. For each course, the standards education goal is given along with details of the modules intended to be used to achieve that goal.

Issues in Information Systems
Volume 19, Issue 3, pp. 131-138, 2018

Computer Ethics Course Outline
Goal: *to establish a baseline of standards as a driving force in computing*

The Online Ethics Course will consist of four modules. Each module has sub-sections. The material is presented as single concepts through short <10 minute videos featuring a professor at Georgia Southern University. This is similar to the method successfully used in Kahn Academy (Khan Academy, 2007). At the end of sub-sections there will be an auto-graded online quiz to tests student's knowledge.

Module I. Standards and their Role
1. Classifications of Standards; An Ontology
2. Standardization Bodies and Processes
3. Trust (including compliance, accreditation etc.)
4. Online Quiz I

Module II. The Basis for Ethical Standards
1. Morality versus Ethics
2. Five Sources of Ethical Standards
3. Online Quiz II

Module III. Professional Ethical Standards
1. ACM Code of Ethics
    a. General Moral Imperatives
    b. Professional Responsibilities
    c. Organizational Leadership Imperatives
    d. Compliance
    e. Online Quiz III-1
2. IEEE/ACM Software Engineering Code of Ethics
    a. Principles 1-8
    b. Online Quiz III-2
3. IEEE Initiative for Ethical Considerations in AI and Autonomous Systems
    a. Principle 1 – Human Benefit
    b. Principle 2 – Responsibility
    c. Principle 3 – Transparency
    d. Principle 4 - Education and Awareness
    e. Online Quiz III-3
4. The Computer Ethics Institutes Ten Commandments of Computer Ethics and the League of Professional System Administrators Code of Ethics
    a. Online Quiz III-4

Module IV. Data Privacy and Protection Standards
1. Data Privacy, Security and US Law
    a. General US Privacy Laws
        i. Fourth Amendment
        ii. US Privacy Act
        iii. Federal Information Security Management Act of 2002
        iv. PATRIOT Act
    b. US Communication Privacy Laws
        i. Electronic Communication Privacy Act (ECPA)
        ii. Communications Assistance for Law Enforcement Act (CALEA)
    c. Information Processing and Security Laws
        i. Uniform Computer Information Transactions Act (UCITA) – model law
        ii. Homeland Security Act (HSA) – Title X (information security)
    d. Financial Privacy Laws
        i. Fair Credit Reporting Act
    e. Medical Privacy Laws

i. HIPPA
f. Online Quiz IV-1

2. The EU and the General Data Protection Directive
   a. European Standards on Confidentiality and Privacy in Healthcare
   b. General Data Protection Directive
      i. Scope
      ii. Responsibility and Accountability
      iii. Lawful Basis for Processing
      iv. Consent
      v. Data Protection Officer
      vi. Pseudonymisation
      vii. Data Breaches
      viii. Sanctions
      ix. Right of Access
      x. Right to Erasure
      xi. Data Portability
      xii. Data Protection by Design and by Default
      xiii. Records of Processing Activities
   c. Online Quiz IV-2
3. Intellectual Property Rights
   a. Copyright
   b. Patents and Trademarks
   c. Determining Fair Use
   d. Digital Millennium Copyright Act (DCMA)
   e. Online Quiz IV-3
4. ISO Data Protection Standards
   a. ISO 27001/27002 – Information Security Management Systems
   b. ISO 27018 –Cloud Privacy Protection
   c. Online Quiz IV-4
5. Federal Information Processing Standards (FIPS)
   a. Security Requirements for Cryptographic Modules
   b. Secure Hash Standards
   c. Digital Signature Standards
   d. Advanced Encryption Standard
   e. Hash Message Authentication Code
   f. Standards for Security Categorization of Federal Information and Information Systems
   g. Minimum Security Standards for Federal Information and Information Systems
   h. Personal Identify Verification of Federal Employees and Contractors
   i. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions
   j. Online Quiz IV-5

System and Software Assurance Course Outline

Systems and Software Assurance is an advanced sophomore level elective designed for students progressing on the software assurance pathway. This course will be prepared for hybrid delivery (supporting alternatively face to face, online, and combined delivery modes) in order to accommodate diverse situations of potential academic and possibly professional users. The online Systems and Software Assurance course consists of four modules, each with several sub- sections generally organized around a lead lecture. The specialized nature of the course and primary assessment approach will require extensive work by the students. Within this course quizzes are scored as a percentage, but not graded (and can be retaken until passed to the student's satisfaction. It is recommended to require that all quizzes are passed with 70% or better.

Module 1: Orientation (2 weeks)

1. Lecture 1: Workforce requirements mapped to certifications
   a. NICE Framework
   b. NICE CWF (NIST SP 800-181) 2017, and NISTIR 8193-draft 2017):
   c. Relevant certifications: CISSP, ISSEP, ISSAP 2017, and CSSLP 2017

   d. M1 Online Quiz-1
2. Lecture 2: Software Assurance Professional Competency Model (DHS, SEI)
   a. Career options: Software Development and System/Software Assurance
   b. M1. Self –Assessment- including taking Myers-Brigs and aptitude tests
   c. M1.Assignment – Breaking into security (two levels career planning, 1-10 years horizon);

Module 2: Security System Defenses and Risk-based Assessment Methodologies (5 weeks)
1. Lecture 1: System Security Principles, Processes and Standards
   a. M2 Online Quiz-1
2. Lecture 2: Controls (exploration of CIS Controls V7 and applicable NIST SP-800 guidelines etc.)
   a. M2 Online Quiz-2
3. Lecture 3: Threats, Assets, Vulnerabilities and Attack Patterns (CVE/CWE, CAPEC)
   a. M2 Online Quiz 3
4. Lecture 4: Risk-based Assessment Methodology: NIST RMF, OCTAVE Alegro, PASTA, ISO 27005
   a. M2 Online Quiz 4
5. M2 Assessment- Security system specification and risk based assessment;

Module 3: Architectural evaluation (3 weeks)
1. Lecture-1: Software System Architecture Evaluation
   a. M3 Online Quiz-1
2. Lecture 2: Threat modeling (standardized DFD representation, using MS TM tool)
   a. M3 Online Quiz-2
   b. M3.Assignment: Design a security application system, completely with a threat model, and perform Architectural Evaluation including considerations for change cases; *Note: for example a realistic logon application for a single sign on (with varying requirements);*

Module 4: Secure Coding (5-weeks)
1. Lecture-1: Secure Software Standards (for example CERT Java, MISRA C)
   a. M4 Online Quiz-1
2. Lecture-2: Enforcing standards by IDE
   a. M4 Online Quiz-2
3. M4 Assignment 1: IDE setup;
4. Lecture- 3: Code M3 security application and perform Code Review
5. M4 Assignment 2: Code the Case from M3, and perform Code Review
6. Lecture 4: Static Analysis (using tools)
   a. M4 Online Quiz-3
7. Lecture-5: Security Testing
   a. M4 Online Quiz-4
   b. M4 Assessment 3: Test plan and test report for testing (breaking) M3 security application; *Note: M4 Assignments ideally should use team work and evaluate work items produced by others*

Software Testing and Quality Assurance outline
Software Testing and Quality Assurance is a senior level elective designed for students progressing on the software assurance pathway. This course will be prepared for online delivery at the three levels outlined earlier in the document. This online Software Testing and Quality Assurance package will consists of three modules, each with several sub-sections and an assessment at the end. The primary assessment approach is an auto-graded quiz, which follows a practice quiz. However, the "apply level" sub-section's assessment will be a scenario based problem for hands-on application of the specific standards to assure higher levels of competency on Blum's scale.

Module 1
1. Overview of a Test Planning Documentation
   a. Quiz I
2. Understanding the IEEE 29119-3/IEEE 829 format
   a. Test Policy and Strategy
   b. Test Plan and Reports

    c. Dynamic Test Process Documentation
    d. Quiz II
3. Using the IEEE 29119-3/IEEE 829 template
    a. Practice Scenario I
    b. Assessment Scenario II

Module 2
1. Overview of Testing techniques
    a. Quiz I
2. Understanding the IEEE 29119-4 format
    a. Specification-Based Testing Techniques
    b. Structure-Based Testing Techniques
    c. Experience-Based Testing Techniques
    d. Quiz II
3. Using IEEE 29119-4
    a. Practice Scenario I
    b. Assessment Scenario II

Module 3
1. Overview of ISTQB certification
    a. Foundation Level
        i. Quiz I
    b. Advanced Level
        i. Quiz II
    c. Expert Level
        i. Quiz III

Software Engineering Capstone Project Course Outline
This course serves as the capstone course for the major and is generally taken in the final semester. This course will be prepared for online delivery and will have elements/options to be assessed at all applicable levels outlined earlier. This online Software Engineering package will consists of three modules, each with several sub-sections and an assessment at the end. The primary assessment approach is an auto-graded quiz, which follows a practice quiz. However, the "apply level" sub-section's assessment will be a scenario based problem for hands-on application of the specific standards to assure higher levels of competency on a Blum's scale.

Module 1
1. Overview of a Software Requirements
    a. Quiz I
2. Understanding the IEEE 29148 format
    a. Test Policy and Strategy
    b. Quiz II
3. Using IEEE 29148
    a. Practice Scenario I
    b. Assessment Scenario II

Module 2
1. Overview of Software Architecture and Software Design
    a. Quiz I
2. Understanding the IEEE 1471 and IEEE 1016 format
    a. Quiz II
3. Using IEEE 1016
    a. Practice Scenario I
    b. Assessment Scenario II

Module 3

1. Overview of Software Project Management
    a. Quiz I
2. Understanding the IEEE 1058 format
    a. Quiz II
3. Using IEEE 1058
    a. Practice Scenario I
    b. Assessment Scenario II

Module 4

1. Work-products and Process Reviews
2. Understanding relevant IEEE and ISO standards, and industry guidelines
    a. Quiz
3. Using relevant standards
    a. Practice Scenario
    b. Assessment Scenario

CONCLUSIONS

This paper presents a preliminary design to improve curricula by promoting secure software development through the application of standards. This design provides not only coverage of discipline specific standards, but also the use of these standards as key course components. The design was tested in the 2017-2018 school year, with the 2018/2019 school year to be used for controlled experimentation. This work is under consideration by NIST for financial support. The results are intended for sharing and are expected to be in public domain. Any suggestions will be appreciated. Future plans involve gradually extending the standards based approach to the entire curriculum i.e. enriching other relevant courses such as introductory programming sequence, Computer and Networks Security, Database Design, Object Oriented Analysis and Design, Human Computer Interaction, Distributed Web Systems Design, and Game Programming.

ACKNOWLEDGEMENTS

This work is supported in part by the National Science Foundation under grant DUE-1525208. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

Amoroso, E. (2018, March). Recent Progress in Software Security. *IEEE Software, 35*(2), 11-13.

IEEE Learning Technology Standards Committee. (2002). *1484.12.1-2002 - IEEE Standard for Learning Object Metadata*. Retrieved May 12, 2018, from https://standards.ieee.org/findstds/standard/1484.12.1-2002.html

Jovanovic, V., & Harris, J. (2016). Systems and Software Assurance — A Model Cyber Security Course. *39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* Opatija, Croatia.

Jovanovic, V., Andres, L., & Sherlund, B. (1999). Use of Software Engineering Standards in Teaching. *4th International Software Engineering Standards Symposium and Forum ISESS'99.* Curitiba, Brazil.

Karachalios, K. (2017). *Expert interview: The importance of industry standardsUnderstanding How Technical Statndards are Made and Maintained.* Retrieved from European Patent Office Organization: https://www.epo.org/news-issues/issues/standards.html

Khan Academy. (2007). (Khan Academy) Retrieved May 11, 2018, from kahnacademy.org

Oltsik, J., McKnight, J., & Gahm, J. (2010). *Assessing Cyber Supply Chain Security Vulnerabilities Within the U.S. Critical Infrastructure.* Enterprise Strategy Group.

Overview of xAPI. (2014). (Rustici Software) Retrieved May 11, 2018, from https://xapi.com/overview/

Sargent, J. J. (2017). *The U.S. science and engineering workforce: Recent, current, and projected employment, wages, and unemployment (CRS Report R43061).* Washington, D.C: Congressional Research Service.

SCORM Explained. (2009). (Rustici Software) Retrieved May 11, 2018, from https://scorm.com/scorm-explained/

Sila. (2018, May 1). *Six Simple Steps to Improve Software Security by Reducing Code Errors*. (Sila) Retrieved April 10, 2018, from https://silasg.com/six-simple-steps-to-improve-software-security-by-reducing-code-errors

STEM-CyLe Team. (2013). *https://stem-cyle.cis.fiu.edu/*. (Florida International IUniversity) Retrieved May 11, 2018, from https://stem-cyle.cis.fiu.edu/

Yuan, X., Yang, L., Jones, B., Yu, H., & Chu, B.-T. (2016). Secure Software Engineering Education: Knowledge Area, Curriculum and Resources. *Journal of Cybersecurity Education, Research and Practice, 2016*(1).

Zatko, S. (2016). Rethinking the Role of Security in Undergraduate Education. *IEEE Security & Privacy, 14*(2), 73-78.