NEW CRITERIA FOR FINAL PROJECTS IN COMPUTER SCIENCE PROGRAMS

Vladimir V. Riabov, Ph.D.[*]
Professor & Department Coordinator, Department of Mathematics & Computer Sciences, Rivier University

Keywords: *computer science curriculum, capstone project, project criteria, course portfolio*

Abstract

The practices of computer-science majors in capstone-project developments are analyzed revealing the lack of novelty, weakness of applications, and low quality of the project-related artifacts. The project criteria have been revised and changed guiding the students to work successfully on the state-of-the-art challenging research projects, to build solid project portfolios, and to go "an extra mile" in their starting careers. The examples of students' outstanding capstone projects (developed within the frame of the revised criteria) are reviewed.

1 INTRODUCTION

The Capstone Project is the last core course (some sort of the Master Theses) in Rivier's Computer Science curriculum. Students prior to the start of the semester individually select topics for their projects and work on the design and implementation of moderately large software systems as the deliverables for this course. At the end of the semester, every student presents the results of her/his project work and the project system demo to the rest of the class.

In the past (before the course revision), "shopping" for "reasonably-good" grades, many students (about 70% of the class) selected basic topics (which are "popular" on the Internet) for their capstone projects, e.g., "Library Online Management System", "Furniture Online Shop", "Online Bus Reservation System", "Mobile Billing System", etc. These "shallow" projects did not reveal neither the student's actual potentials nor their readiness to deal with the career challenges of applying the state-of-the-arts technologies in the high-tech company projects.

Many of these students ("grade shoppers") came from abroad (mostly from India and China) and brought new challenges associated with the cultural and educational-system differences, such as weak academic background; weak code-programming and communication skills; ineffective study skills; numerous cases of cheating and plagiarizing (practically undetected among our "traditional" local graduate students) [1, 2], and attempts to avoid challenges in research studies by searching and adapting the existing patterns of the problem solving.

Several years ago, Rivier's CS/IT faculty members had reviewed the practices of running the Capstone Project course, revised the course criteria and even renamed the course to the Professional Seminar. Course instructors start promoting the method of critical thinking in their teaching practices. The method is based on independent research approach, challenging topics, searching for non-trivial solutions, fair discussions, and respect for different opinions through an overview and reflections on opposite arguments. The class discussions have started covering various modern societal issues, e.g., "Environments Become Smart", "Life after the Internet", "How Biology Became an Information System," "Environment in Human-Centered Systems," "Engineering the Ocean," and others [3, 4].

ISSN 1559-9388 (online version), ISSN 1559-9396 (CD-ROM version).

Every student has prepared for and led discussions on two selected peer-reviewed articles from professional journals [5, 6], magazines, and/or books on the future of computing. There are a few examples of the successful students' survey reviews: "A New Experimental Website Converts Photos into 3D Models", "Taking Measure of SaaS Reliability", "Software Reliability Methods and Experience", "An Overview of Malware", "Top Ten Computer Viruses All the Time", "Automatic Information Extraction from Large Websites", "Online Analytical Processing (OLAP)", "Computers 'Taught' to Search for Photos Based on Their Contents", "Quantum Computing", "The Adobe™ AIR Audio Player", "Flash Drives: Latest and Greatest Gadget", "The Future of Electronic Displays", and others. This search for knowledge helps the students stay at the cutting edge of computer science.

In addition to these activities, students read all chapters from the *Craft of Research* textbook [7] that helps them develop research skills for their work on the individual capstone projects.

In this paper, the revised course criteria (such as the orientation on practical application, novelty, the structured methodology of project development, quality of the project-related artifacts, creating a system demo, and building a project portfolio), the "extra mile" opportunities, and examples of students' outstanding CS capstone projects are considered in details.

2 REVISING THE COURSE CRITERIA

2.1 Focusing on Practical Applications

The selection of a project topic is the first step in the project development. Students are encouraged to conduct the feasibility analysis of the system potential users and the expected valuable services provided by the proposed system to the users. The students, who have experience of working in high-tech companies or taking internships there, have typically selected the interesting, challenging project topics promising strong practical applications. The others struggle with the project proposals, requesting extra two-three weeks (after their overviews of a few research articles) to find the appropriate project topics.

Finally, a course instructor approves the project proposals or proposal revisions. In the past, the projects covered many important practical applications, including "Portable Graphical User Interface (GUI) in Ada and GtkAda", "Distributed Backup System", "Visual Data Structures", "A New Chat System", "Simulation of Prefetching the Web Access Streams", "User-friendly Database Development in PERL", "Windows™ Covering System with Oracle™ Database Server", "Designing the Blade Servers", "Building a Data Warehouse", "Struts for Building a Web Application with Java", "Qualitative Analysis of the Practical Applications of Open Source Software (OSS) and the Software as a Service (SaaS) Model", "An Extensible Application Framework using LINQ and XML", "Driver's License Tracking System", "Exploring LISP and Ruby: Case Study of a LISP Interpreter", "The StructureScan Mini Viewer for Ground Penetrating RADAR Data", and many others.

2.2 Novelty

Natural curiosity and novelty bring the talented students the greatest ecstasy that governs them in discovery endeavors in their academic lives and, later, in professional careers. The students have tried to introduce new elements in their capstone projects and got the outstanding results. The novelty has featured in the wide variety of students' projects, including "Modeling a Digital Video Cluster", "Personal Encrypted Talk Tool", "A Public Resource Computing Platform for Simulating N-Body Galaxies", "A Multi-Domain Musician's Web-service Using Ruby-on-Rails, SOAP, FLEX, and AJAX", “Technique for InfiniBand™ Dynamic Memory Registration”, “Secure Online Biometric Authentication Solution”, “System Reliability Calculator Tool”, “Voice-over-IP Call Management and Networking Utility Tool”, “Visualization of Multivariate Data through Chernoff Faces”, “Clue Computer Game”, “Tactus: A World Learning Game for Children with Autism”, “Design and Implementation of an IoT Smart Farming System”, “DEM846: Digital Elevation Modeling”, and others. Some of these projects are reviewed in the special section of this paper below.

2.3 Project Development Stages and Standards

As the general requirement for this course, students should follow the established project-development procedures (“stages” of project planning, feasibility and functional analyses, system design, code programming and implementation, and system prototype testing) [8] and standards [9, 10]. Typically, this process takes 12-14 weeks. Prior to these activities, students develop mini projects on the elements of the small system development in various core and elective CS courses, including the Object-Oriented System Analysis & Design, Computer Architecture, Operating Systems, Software Engineering, Database Management Systems, Multimedia & Web Development, Computer Security, Java Programming, C/C++ Programming, Exploring Perl & Ruby, and Software Quality Assurance. The acquired knowledge and skills have been successfully used by the students in their work on the capstone projects.

2.4 Quality of the Project-Related Artifacts

In the project evaluations, the course instructors draw special attention to the quality of various project-related artifacts, e.g., in the case of the object-orient system development approach [9], students are required to produce the activity diagram, use case diagrams (with the detail description of external actors/users/roles and the associated use cases), class and object diagrams (related to every use case), behavioral sequence, communication and state-machine diagrams [11], package diagrams (where software packages are populated with classes) for various system architectural levels, deployment diagrams (built in the Client-Server approach with the proper allocation of software components [presentation, application, data management, and data storage layers] on all clients and servers), user-computer interfaces, windows navigation diagrams, normalized data tables, entity-relational diagrams, and codes written in various programming languages. In the case of the procedural system development approach [8, 12], the appropriate activity and data flow diagrams should be produced.

After every stage of the system development process, the corresponding test plans have to be created, e.g., the Functional Test Plan, the System Integration Test Plan, the Unit/Module Test Plan, and, finally, the Device Verification Test Plan. The Structured Testing Methodology (based on the system complexity analysis with McCabe’s metrics) [13, 14] is used for the test planning and test running. The test and code coverages are also investigated [15]. The special attention is paid to the estimation of the “projected” code errors using the Halstead’s metrics [16].

The special requirements have been developed for the high quality of the project reporting (the style of project report organization, grammar and spell checks, citation of sources, image quality, etc.), as well as for the project oral presentations. These efforts help students develop strong “soft skills” that have become in the growing demand in the companies and in the scholar community.

2.5 The System Demo Requirement

Every student delivers publicly the project presentation with demonstration of the system prototype (as the initial “Engineering Release”). Prior to their presentations, students are encouraged to set up a proper computing environment for running the system demo. Typically, they use their laptops or the remotely allocated servers for the project demonstration. Nobody expects that the system performance will be perfect, but the student has to be capable to explain why the certain errors occur. This is a normal situation of trials and errors associated with the real system development process.

2.6 The Project Portfolio

Finally, at the end of the course, students are required to submit the project portfolios with all the project documentation (including a project report, code and database files, system environment settings, user’s manual, and other project-related artifacts) burnt on a CD or accessible from the student’s personalized website. In many cases, the students also use these portfolios during the internship or job interviews and for the job promotions in the companies.

3 EXAMPLES OF FINAL CAPSTONE PROJECTS

Many students selected challenging topics for their capstone projects. Here we only make overviews of a few outstanding projects that have been performed using the advanced computational concepts.

David Snogles developed the Personal Encrypted Talk system for his final capstone project. Its primary goal was to secure Instant Messaging communications between two parties on the Internet. Secondary objectives were Java Cryptography Architecture research and the practical experience gained by the student in the development of a scalable Java-based Graphical User Interface application. His article [17] summarizes the software engineering steps followed during the project implementation.

In his project “Designing an Easily Modifiable Cipher for Educational Purposes” [18], Michael Jeffords introduced two easily-implemented ciphers that can be used to teach the basics of crypto-analysis using the character frequency analysis. In this approach, each cipher builds from the previous one, and the implementer can choose to keep contextual clues such as spacing and capitalization. These ciphers were designed to both hide letter frequencies and to be broken easily. The goal was that this approach could be used to excite students about the field of crypto-analysis and steer them toward open standards of encryption. Michael explored the ideas of the Vernam cipher [19] and worked on its implementation with the RC4 algorithm [20].

In his research project, Douglas Selent used the McCabe IQ™ tool to analyze source code complexity of the Light-up Puzzle program [21] that he created. The source code was converted into a graph, which then was used to analyze the source code complexity. The features of the McCabe™ IQ tool that were used in this project are the Battlemap, System Complexity metrics, method flowgraphs, scatter diagrams, Halstead metrics, Class metrics, and Object-Oriented metrics. The tool allowed him to determine if any parts of the source code are unreliable or unmaintainable, and make proper code corrections. This approach helped him identify vulnerable code areas, reduce error rates, shorten testing cycles, improve maintainability, and maximize reusability. In order to verify the effectiveness of the McCabe™ IQ tool, he re-factored the program in the areas reported to be highly complex and error-prone. After this, he compared the McCabe’s Metrics reports on the initial analysis to the reports on the re-factored analysis and charted results to clearly indicate the improvements made to decrease complexity. This technique improved significantly the quality of his software code.

Recent examples of outstanding projects of students in the Computer Science Professional Seminar include the following: Chaitrali Jayantilal Doshi applied the Smart Grasses technology (Xamarin™ Cross Platform framework explored during her internship with Tego, Inc.) for developing an App application (for Android, iOS, and Windows operating systems) that allows a user to adapt the Smart Glasses (Google™ Glasses or Vuzix™ Smart Glasses) to augment reality; Praveen Kalyan Indrakanti developed the secure cloud storage system (Searchable Cryptosystem) with smooth projective hash functionality that allows using open key encryption with watchword tests in numerous utilizations of distributed storage; Sruthartha Cheedu developed a Facebook™ Chatbot system with Amazon™ Lex capability of conversations through textual messages between the system users; Dharma Teja Adapa developed the secure cloud storage system that preserves user's privacy applying the data mining approach with 3D-RBT perturbation technique; and Xiaoxu Gu explored modern web development frameworks (Angular™ and Spring™) to create a personal accounting WebApp.

4 GOING AN EXTRA MILE

The method of encouraging students to go an extra mile and challenge the opportunities helps the students in building-up the faith in their abilities and leadership qualities, which are in the great demand by the society (in industry, business, government, and community) nowadays. Faculty encourage the students to pursue scholarly activities (e.g., publish the first article in a peer-reviewed research journal, or present a paper at a conference), support the intellectual growth of the students, and share their research results with the global community of scholars. As an example, Saroj Maharjan applied in his capstone project the Struts™ methodology framework for developing a web application that allows a user to adapt to the social networking environment (his article based on the summary of this capstone project was later published in [22]).

Vijaya Dommeti, a MS/CS student, and her course instructor and mentor Prof. Douglas Selent, presented the paper "Applying and Exploring Bayesian Hypothesis Testing for Large Scale Experimentation in Online Tutoring Systems" at the poster section of the ACM Conference that was held at the Massachusetts Institute of Technology in Cambridge, MA, in April 2017. Their manuscript was recently published in the Proceedings of the ACM Conference [23].

Kevin Gill developed the Living Mars image project [24] that included topics related to computer graphics, software development, astronomy, and planetary science. The purpose of the project was to create a visualization of the planet Mars as could look with a living biosphere. This makes no distinction as to whether this biosphere would represent an ancient or future, possibly terraformed planet. The algorithms and methods used in generating shadows on digital elevation models were developed in his previous study [25]. These include formulas that are common in computer graphics applications and are often provided by specific frameworks (i.e., OpenGL). The basics of model rendering are covered from the structure of the source data to the interpolation of hypsometric/bathymetric tint colors. The primary algorithm presented in [25] is the calculation of shadows using ray tracing. The methods are based on the code from the jDem846 open source project managed by the student. Code listings are provided in [25] and available in full online. Kevin's studies were reviewed by NASA specialists, and later he was invited to join their team and work on several space missions at the Jet Propulsion Laboratory.

Several talented students included their capstone-project portfolios into applications for the further studies in Ph.D./Computer Science programs at M.I.T., W.P.I., UMASS-Lowell, UNH, and other universities. Upon completing successfully these programs, they continue collaborating with our department in many ways, including the lecturing, the supervising of our interns, and the students' mentoring.

5 CONCLUDING REMARKS

The revised criteria for capstone projects in M.S./CS program have guided students to focus on valuable practical applications, demonstrate novelty in adapting modern computing technologies, apply the structured methodology of project development, achieve the high quality of the project-related artifacts, create a robust system demo, and build a solid project portfolio. The examples of students' capstone projects demonstrate their successful works on the state-of-the-art challenging research and applications. Promoting their capstone-project achievements and portfolios, many students also select the "extra mile" opportunities applying for studies in Ph.D./Computer Science programs in other universities, presenting papers at national and international conferences, and publishing their first research articles in professional journals. This demonstration of the acquired skills and the quality of their project-related work and research helps the students to pass successfully the job and/or internship interviews or get a promotion in high-tech companies.

In the course evaluations, students stated that they became deeply engaged in capstone-project activities through examining the challenging problems related to the real-world applications of the modern computing technologies.

REFERENCES

[1] Ritesh Chugh. Students are using 'smart' spy technology to cheat in exams. On *www.sciencealert.com*, May 17, 2016. Retrieved October 28, 2019, from http://www.sciencealert.com/students-are-using-smart-spy-technology-to-cheat-in-exams

[2] Koh Gui Qing, et al. How an industry helps Chinese students cheat their way into and through U.S. colleges. *Reuters*, May 25, 2016, 6:30 p.m. GMT. Retrieved October 28, 2019, from http://www.reuters.com/investigates/special-report/college-cheating-iowa/

[3] P. J. Denning and R. M. Metcalfe, *eds. Beyond Calculation*. New York, NY: Fall-Verlag, 1997.

[4] P. J. Denning, *editor. The Invisible Future*. New York, NY: McGraw-Hill, 2002.

[5] The ACM Digital Library. Association for Computing Machinery. Online: https://dl.acm.org/

[6] The IEEE Computer Society Digital Library (CSDL). Institute of Electrical and Electronics Engineers. Online: https://www.computer.org/csdl/home

[7] W. C. Booth, G. G. Colomb, and J. M. Williams. *The Craft of Research*, 4th edition. Chicago, IL: The University of Chicago Press, 2013.

[8] R. S. Pressman. *Software Engineering: A Practitioner's Approach*, 4th edition. New York, NY: McGraw-Hill, 1997.

[9] A. Dennis, B. H. Wixom, and D. Tegarden. *System Analysis and Design: An Object-Oriented Approach with UML*, 5th edition. New York, NY: John Wiley & Sons, Inc., 2015.

[10] S. R. Schach. *An Introduction to Object-Oriented Systems Analysis and Design with UML and the Unified Process*. New York, NY: McGraw-Hill, 2004.

[11] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Reading, MA: Addison Wesley, 1999.

[12] L. Fertuck. *System Analysis and Design with Modern Methods*, 2nd edition. New York, NY: William C. Brown Communications, Inc., 1994.

[13] T. J. McCabe. A complexity measure. *IEEE Transactions on Software Engineering*, SE-2(4):308–320, 1976.

[14] A. H. Watson and T. J. McCabe. Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric. *NIST Special Publication*, No. 500-235. Gaithersburg, MD: National Institute of Standards and Technology, September 1996.

[15] V. V. Riabov. Methodologies and Tools for the Software Quality Assurance Course. *Journal of Computing Sciences in Colleges* 26(6), 86-92, 2011.

[16] M. H. Halstead, *Elements of Software Science*. New York, NY: Elsevier North Holland, 1977.

[17] D. Snogles. Personal Encrypted Talk - Securing Instant Messaging with a Java Application. *Rivier Academic Journal* 1(1), 1-13, 2005. [Online] https://www2.rivier.edu/journal/ROAJ-2005-Fall/J14-SNOGLES.pdf

[18] M. Jeffords. Designing an Easily Modifiable Cipher for Educational Purposes. *Rivier Academic Journal* 8(1), 1-10, 2012. [Online] https://www2.rivier.edu/journal/ROAJ-Spring-2012/J603-Jeffords.pdf

[19] Wikipedia, *Gilbert Vernam*. [Online] http://en.wikipedia.org/wiki/Gilbert_Vernam/

[20] RSA Laboratories. *What is RC4?* [Online] http://www.rsa.com/rsalabs/node.asp?id=2250

[21] D. Selent. The Design and Complexity Analysis of the Light-Up Puzzle Program. *Journal of Computing Sciences in Colleges* 26(6), 187-189, 2011.

[22] S. Maharjan. Social Networking Web Application Using Struts Framework. *Rivier Academic Journal* 14(1): 1-19, 2018. [Online] https://www2.rivier.edu/journal/ROAJ-Fall-2018/J1062_Maharjan.pdf

[23] V. Dommeti and D. Selent. Applying and Exploring Bayesian Hypothesis Testing for Large Scale Experimentation in Online Tutoring Systems. In *Proceedings of the Fourth (2017) ACM Conference on Learning@Scale* (Cambridge, MA, April 20-21, 2017), pp. 229-232. doi>10.1145/3051457.3053992

[24] K. M. Gill. Putting Life on Mars: Using Computer Graphics to Render a Living Mars. *Rivier Academic Journal* 9(1): 1-27, 2013. [Online] https://www2.rivier.edu/journal/ROAJ-Spring-2013/J754_Kevin-Gill_Living-Mars.pdf

[25] K. M. Gill. Casting Shadows: Shading Digital Elevation Models Using Ray Tracing. *Rivier Academic Journal* 8(1): 1-16, 2012. [Online] https://www2.rivier.edu/journal/ROAJ-Spring-2012/J614-Gill.pdf

[*] **Dr. VLADIMIR V. RIABOV**, Professor of Computer Science and Department Coordinator at Rivier University, teaches algorithms, networking technologies, computer security, software engineering, software quality assurance, object-oriented system design, system simulation and modeling, numerical methods, introduction to computing, and professional seminar in computer science. He received a Ph.D. in Applied Mathematics and Physics from Moscow Institute of Physics and Technology and M.S. in Computer Information Systems from Southern New Hampshire University. Vladimir published about 130 articles in encyclopedias, handbooks, journals, and international and national conference proceedings, including *The Internet Encyclopedia*, *The Handbook of Information Security*, *The Handbook of Computer Networks*, *International Journal of Computers and Structures*, *Journal of Spacecraft and Rockets*, *Journal of Aircraft*, *Journal of Thermophysics and Heat Transfer*, *Congress Proceedings of International Council of the Aeronautical Sciences*, *International Symposia on Rarefied Gas Dynamics and Shock Waves*, *International Conferences on Computer Science and Information Systems*, *International Conferences on Technology in Collegiate Mathematics*, *Conferences of American Institute of Aeronautics and Astronautics*, and others. He is a senior member of ACM and AIAA, and a member of IEEE and MAA.

Download latest collection of java projects for final year computer science students along with ppt,pdf,project report,source code and abstract for free of cost. Computer science final year students can use these projects as major and minor projects with few modifications. Projects listed here are developed by previous year cse students. submit java projects for final year computer science to us. Links to download java projects for final year computer science: Mobile banking project source code in java.