

## Assessing BS–CS Student Outcomes Using Senior Project

### Mr. Norman Pestaina, Florida International University

Mr. Norman Pestaina is a Senior Instructor in the School of Computing and Information Sciences (SCIS) at Florida International University (FIU). Mr. Pestaina completed the B.Sc. in Mathematics (Special) at the University of the West Indies in 1972, and the MS in Computer Science at the Pennsylvania State University in 1979. He has been an Assistant Staff member of the Massachusetts Institute of Technology's Lincoln Laboratory, and Lecturer in the Department of Mathematics at the Cave Hill campus of the University of the West Indies in his native Barbados, W.I.

A member of the FIU-SCIS faculty for more than 30 years, Mr. Pestaina has taught at all levels of the curriculum, receiving awards for excellence in teaching on five occasions. He served the School as an Undergraduate Advisor for 15 of those years, and has served continuously as a member of the School's Curriculum/Undergraduate Committees. Mr. Pestaina was a principal architect of the School's program assessment processes, and the SCIS undergraduate program Assessment Coordinator from 2006 through 2013, leading successful re-accreditation of the BS in Computer Science program in 2004, and 2010. Mr. Pestaina has been a Reader and Question Leader of the College Board's Advanced Placement Computer Science Exam since 2000.

### Ms. Tiana Solis, Florida International University

Tiana Solis is currently a Senior Academic Adviser /Team Leader at the School of Computing and Information Sciences, Florida International University. Prior to moving to Hawaii in 2007, she was an instructor and academic advisor for the School from 1994 to 2007. Ms. Solis taught different undergraduate courses and mentored several FIU students participating in the Florida-Georgia Louis Stokes Alliance for Minority Participation (FGLSAMP). She is a past adviser of the Women in Computer Science (WICS) student club. From 2008 to 2010. Ms. Solis was a programmer analyst at the Department of the Attorney General in Hawaii, a member of the team revamping the State Juvenile Justice Information System. Her research and instructional interests include software development, programming languages, and computer ethics.

### Dr. Peter J. Clarke, Florida International University

Peter J. Clarke received his BSc. degree in Computer Science and Mathematics from the University of the West Indies (Cave Hill) in 1987, MS degree from SUNY Binghamton University in 1996 and PhD in Computer Science from Clemson University in 2003. His research interests are in the areas of software testing, software metrics, model-driven software development, domain-specific modeling languages, and computer science education. He is currently an associate professor in the School of Computing and Information Sciences at Florida International University. He is a member of the ACM (SIGSOFT, SIGCSE, and SIGAPP); IEEE Computer Society; and a member of the Association for Software Testing (AST).

# Assessing BS–CS Student Outcomes Using Senior Project

## Abstract

Undergraduate program assessment is undertaken by many colleges world-wide in support of their continuous improvement processes. In addition to assuring stakeholders of program quality, assessment is required by major regional and national accrediting agencies. A critical part of the assessment process is the generation of useful data for analysis and evaluation yielding indicators for program improvement. Senior year capstone projects are a fertile source of such data. In this paper, we outline the Student Outcomes and Senior Project course of the BS-CS program at Florida International University (FIU). We describe and evaluate a methodology used to perform assessment of attainment of the BS-CS Student Outcomes using data from the Senior Project course.

## 1. Introduction

Undergraduate program assessment continues to be a significant undertaking in many institutions. It is the critical component of the continuous improvement process, and may provide stakeholders with some confidence in the quality of the programs<sup>3, 4</sup>. Also, assessment and continuous improvement practices are invariably prerequisites for program accreditation, a designation conferred by entities external to the departments in which the undergraduate programs are offered. The Bachelor of Science in Computer Science (BS-CS) program at Florida International University (FIU)<sup>[11]</sup> is externally accredited by both the Computing Accreditation Commission of the Accreditation Board of Engineering and Technology (ABET/CAC)<sup>1</sup>, an international accreditation body, and the Southern Association of Colleges and Schools (SACS)<sup>12</sup>, a regional accreditation body.

The BS-CS at FIU is a dynamic program that implements many of the recommendations outlined in successive ACM/IEEE Computing Curricula for Computer Science<sup>2</sup>. Resulting from an internal review of the BS-CS program undertaken in 2006, the Senior Project course became a required component of the BS-CS program in Fall 2007. Since the BS-CS program structure is topic-based, the Senior Project course was designed as a capstone experience. It enables students to integrate knowledge units from several CS courses such as data structures, database management, operating systems, etc. in a holistic way. The Senior Project also allows students to complete a significant project encompassing both design and implementation, and requiring elements of professional practice such as teamwork, presentation skills and ethical considerations. Introduction of the Senior Project into the BS-CS curriculum also meant that there would now be an excellent source of direct summative assessment information.

The contributions of the paper are:

1. To present our approach to assessing attainment of the BS-CS Student Outcomes based on the Senior Project course.
2. To present an evaluation of the approach using data collected from its application over two years.

The paper is organized as follows: Section 2 presents the related work; Section 3 an overview of the BS-CS program; Section 4 our approach for assessment; Section 5 results of a case study; and Section 6 the conclusion.

## 2. Related Work

In this section we describe the research most closely related to the work presented in this paper.

### 2.1 Assessing BS-CS Programs

Sanders et al.<sup>10</sup> investigated the tools that were being used to assess ABET/CAC accredited CS programs in the US. They found that ten (10) different tools were used including senior exit surveys, alumni surveys, employer surveys, written exams and portfolios (maintained by the department and the students), among others. Bailie et al.<sup>6</sup> describe how rubrics for two programming courses can be used to measure the student learning outcomes derived from the ABET's Program Outcomes. In addition the authors describe how the data collected was used to improve the process.

In our work we show how to use the senior capstone project to evaluate attainment of Student Outcomes (SOs) through the BS-CS curriculum. Our approach assumes a mapping of our SOs to the ABET/CAC 2010 Student Outcomes<sup>1</sup>, and evaluates them by application of a rubric. However, our rubric has a different structure to the one presented by Bailie et al.<sup>6</sup>.

### 2.2 Assessment of Senior Projects

Clear et al.<sup>7</sup> present a report that assists instructors to design, implement and assess capstone courses. The report is a valuable resource for departments starting a new capstone course. Richards<sup>9</sup> describes the key design choices of a project-based course, focusing on the composition of the groups and the issues surrounding assessment. Both Clear et al.<sup>7</sup> and Richards<sup>9</sup> provide comprehensive descriptions of the assessment of projects but there is no description of how the capstone project may be used to assess the SOs of a computer science program. Farrell et al.<sup>8</sup> describe an approach that attempts to develop a system for the fair allocation of course grades to the members of the senior project team. In grading our senior projects we use some of the ideas presented by Farrell et al.<sup>8</sup>, e.g., peer group assessment and evaluating meeting minutes

Ahmad et al.<sup>5</sup> performed a study of the undergraduate software capstone project at 19 Pakistani universities and provides generic support for quality assessment of capstone projects at the undergraduate level. The study investigated the current practices followed for assessment of computer science and software engineering capstone projects and the formulation of generic rubrics for quality assessment to minimize variation in quality.

### 3. BS Computer Science Program

In this section we describe the SOs for the BS-CS program at FIU, identify the principal enabling courses in the program, and describe the rationale for our Senior Project course.

#### 3.1 The BS-CS Student Outcomes

The Student Outcomes (SOs) of an academic program are statements of the general characteristics of the program's graduates. Typically, they express abilities of graduates that are enabled by students' progress through the program. Additionally the SOs are intended to make the broader Program Objectives realizable, and are often adjusted as the Program Objectives evolve.

<p>To complete the program of study for the BS-CS, every student will</p> <ol style="list-style-type: none"><li>Demonstrate proficiency in the foundation areas of Computer Science including discrete structures, logic and the theory of algorithms.</li><li>Demonstrate proficiency in various areas of Computer Science including data structures and algorithms, concepts of programming languages and computer systems.</li><li>Demonstrate proficiency in problem solving and application of software engineering techniques.</li><li>Demonstrate mastery of at least one modern programming language and proficiency in at least one other.</li><li>Demonstrate understanding of the social and ethical concerns of the practicing computer scientist.</li><li>Demonstrate the ability to work cooperatively in teams.</li><li>Demonstrate effective communication skills.</li><li>Have experience with contemporary environments and tools necessary for the practice of computing.</li></ol>
--

Figure 1: FIU BS-CS Student Outcomes

The current iteration represented in Figure 1 is already scheduled for update during the 2012-13 academic year.

#### 3.2 Enabling Courses

The principal enabler of the SOs is the curriculum. For the BS-CS at FIU, the required courses most closely aligned with each SO are listed:

- Foundations: Discrete Mathematics, Logic for CS, Theory of Algorithms;
- CS Core: Data Structures, Principles of Programming Languages, Database Management, Computer Organization, Operating Systems Principles, required CS electives;
- Software Development: Software Engineering I, Senior Project;
- Programming: Computer Programming I, II & III;
- Ethics: Ethics and Social Issues in Computing;
- Teamwork: Software Engineering I, Senior Project;

- g) Communication: Professional and Technical Writing; Business and Professional Communication; Senior Project.
- h) Computing Tools: Operating Systems Principles, Database Management, Software Engineering I, Senior Project.

It should be noted that a recent BS-CS curriculum review has resulted in several changes effective in Fall 2012.

### 3.3 Senior Project

Our Senior Project course CIS 4911 was designed as a capstone experience for our graduating seniors. It is a required course of the BS in CS curriculum:

**Catalog Description:** *Students work on faculty supervised projects in teams of up to 5 members to design and implement solutions to problems utilizing knowledge obtained across the spectrum of Computer Science courses. This course should be taken during the semester in which the student completes all the CS courses required for the CS major.*

Figure 2: Catalog description of CIS 4911 Senior Project

Figure 2 shows the catalog description of CIS4911, the complete operational CIS 4911 syllabus may be viewed from: [http://www.cis.fiu.edu/programs/undergrad/courses/CIS\\_4911.pdf](http://www.cis.fiu.edu/programs/undergrad/courses/CIS_4911.pdf)

Our BS-CS program is organized following a traditional topic-based approach. The knowledge units are delivered in disjoint subject-focused courses: data structures, database management, operating systems, etc. Although there is a prerequisite structure, the intersection of knowledge units between courses is minimal. It is now generally accepted that successful practice in a computing field requires a holistic competence. One role of the capstone course in a topic-based curriculum is to foster an appreciation of how the knowledge units relate, and to forge that holistic competence. Inclusion of a capstone course into a topic-based curriculum is recommended in ACM Curricula 2001<sup>2</sup>.

Currently, two flavors of the Senior Project are offered; one with a software engineering focus and the other with a systems focus. Both require development of a software system and include requirements specification, design, implementation and some form of validation. Students are required to submit four (4) deliverables during the semester and make an oral presentation of each deliverable. All presentations are recorded so that students and instructors can review the presentation performances.

## 4. Approach for Assessment

In an educational context, assessment is a process for gathering information about elements of an educational program, analyzing the information to extract indicators of program effectiveness, and evaluating the indicators against program expectations: i) to make inferences about program effectiveness, ii) to identify signposts for improving the program, iii) to make adjustments to the assessment process itself. The SOs are one set of program markers. Precisely because of the

holistic nature of a well-designed capstone, all SOs can reasonably be expected to be manifested via completion of a capstone project. This confluence accounts for the suitability of the capstone or Senior Project course as a medium for assessment of the program.

Our BS-CS program utilizes assessment of SOs in the Senior Project course as one component of its assessment plan. Other components include course-embedded assessment via quizzes and application of specialized rubrics to course artifacts; in addition, indirect measures are obtained by surveying students, instructors, alumni and employers. In the remainder of this section, the structure of the Senior Project assessment process is described.

#### 4.1 Data Collection

One source of assessment raw data is the set of completed projects – presentations and artifacts. The data are extracted by applying rubrics to the presentations and artifacts. The agents applying the rubrics are evaluators, faculty members or industry clients, and student team-members.

Unsurprisingly, the master rubric, *Senior Project Rubric*<sup>11</sup>, has eight sections, one for each of the eight Student Outcomes, see Section 3.2:

- The first five sections, (a) Foundations, (b) CS Core, (c) Software Development, (d) Programming, and (e) Ethics are check-lists completed by the evaluators. In each section, the sum of the check-marks (1 if checked, 0 if not checked), up to a limit of 5, provides a rating for attainment of the Student Outcome of that section. See the rubric for the Foundations Student Outcome in Appendix A.
- The principal component of section (f) Teamwork is the *Teamwork Peer Rating Rubric*<sup>11</sup>. It is completed by each team-member to rate participation of each of their other team members. Five team-work facets are rated against two criteria each. These ten ratings provided by each team-member are averaged to obtain ten criterion team ratings in the range 1 to 5. Each team-work facet earns a check-mark only if both of its criteria receive an average rating of at least 4. The evaluators provide check-marks for two other team-work facets. The sum of the check-marks, up to a limit of 5, provides a rating of the Teamwork Student Outcome. See the rubric for the Teamwork Student Outcome f) in Appendix B.
- The principal component of section (g) Communication is the *Presentation-Skills Rubric*. It is completed by the evaluators to rate the oral presentation skills of each team member. Each presenter is rated on five presentation facets using traditional criteria. For each facet, the presenters' ratings are averaged to obtain a team rating for that facet. The facet earns a check-mark if the team rating is at least 3 from a maximum of 4. The evaluators provide check-marks for two other facets. The sum of the check-marks, up to a limit of 5, provides a rating of the Communication Student Outcome.
- Section (h) Computing Tools is entirely a *Computing Tools Rubric* where each project team, collectively, enumerates the tools and environments employed by the team, and provides self-ratings of their proficiency with each environment or tool. The tools are categorized by application domain: Modeling, Project Management, DBMS, etc. The domain earns a check-mark if the team's self-rating of competence with the domain tools is above the *novice* level.

The sum of the check-marks, up to a limit of 5, provides a rating of the Computing Tools Student Outcome.

At the top level, the 1<sup>st</sup> Tier, the methodology is enumerative; it checks for utilization of SO knowledge area facets in the project implementation. A lower level, the 2<sup>nd</sup> Tier, provides more fine-grained information that might be useful for focusing continuous improvement efforts.

#### 4.2 Analysis

Each Senior Project is independently rated by two evaluators via application of the *Senior Project Rubric*<sup>11</sup>. This yields two ratings of attainment of each SO in the range 1 to 5. The ratings are discrete numbers. The paired ratings are combined to obtain a single project-rating for each SO:

- When the paired ratings are identical, the common rating is the SO rating.
- When the paired ratings differ by 1, their average is the SO rating.
- When the paired ratings differ by more than 1, a mediated rating is provided by a third evaluator (mediator).

For each SO, the project-rating from all Senior Projects in the semester are averaged to obtain a semester rating of attainment of the SOs. These ratings are reported in the semester summary of direct assessment data. The data from all semesters in an academic year are reported in an annual summary.

#### 4.3 Evaluation

There is frequent misunderstanding of what outcomes assessment is about; that is, what is being evaluated. In this instance, it is not the students, or their projects, or the quality of mentorship, or the curriculum. What is being evaluated is the efficacy of the program; the assessment process described here seeks to answer the question “*How well does the BS-CS program enable its graduates to attain the program’s SOs*”. The data analysis yields metrics that are interpreted as answers to this question for each SO.

To evaluate using metrics, a *standard* or minimal acceptable rating value must be established. The default standard adopted by our program, for all measures, is 75% of the maximum rating. In this case the standard is 3.75; that is the threshold at which the Senior Project Student Outcome ratings are deemed acceptable, and below which a rating raises a red flag.

Example: the AY2011-12 summary reports Senior Project ratings of **4.56** (91%) for SO (b), and **2.72** (54%) for SO (e). Taken alone, these suggest good attainment of the SO (b), but poor attainment of SO (e). In practice, each SO is evaluated using several metrics derived via course-embedded assessment as well as surveys. There are good reasons for this seeming duplication of effort:

- Having multiple indicators increases the potential for meaningful evaluation.
- Especially for continuous improvement purposes, it is helpful to have both formative and summative indicators.

- Some assessment methods may be poorly suited to evaluating a particular outcome.
- A particular assessment instrument (rubric, survey, and quiz) may be poorly designed, or awkward to execute.
- An agent executing a particular assessment instrument may do so carelessly, or with intent to mislead.

In the particular case of our *Senior Project Rubric*<sup>11</sup>, any questions of applicability and consistency should be resolved in order to have confidence in the metrics it provides. It must also be flexible enough to accommodate modifications or additions of SOs. The timing of this study is, in part, dictated by impending changes to the BS-CS SOs for closer alignment with the ABET SOs.

## 5. Case Study

As described earlier, CIS 4911 is a capstone course coming towards the end of the students' program of study. This makes CIS 4911 a prime source for summative assessment of attainment of SOs, and accounts for the significant effort invested in development of the *Senior Project Rubric*<sup>11</sup>.

### 5.1 Research Questions

The overarching question is whether to adapt the existing *Senior Project Rubric*, with some (hoped) improvements, to a revised set of SOs. To that end, some desired attributes of the *Senior Project Rubric* are evaluated:

RQ1 (*Applicability*): Does the rubric's methodology measure attainment of all of the SOs in a meaningful way? If not all, to which SOs is it not applicable?

RQ2 (*Extendibility*): Is the rubric methodology(s) applicable to varied SO categories, and does it lend itself to fine tuning?

RQ3 (*Consistency*): Is there reasonable expectation that repeated application of the rubric to identical data should yield identical metrics? When there is not, can this shortcoming be ameliorated?

### 5.2 Method

The sources of the data for this study are 22 capstone projects completed between Fall 2010 and Spring 2012, encompassing the efforts of about 82 students. The numeric data were obtained by applying the *Senior Project Rubric* described earlier to the presentations and artifacts of these 22 projects. Informal reactions and feedback from evaluators applying the *Senior Project Rubric* provide useful insight. Thus, some anecdotal experiential information is utilized.

For each SO, evaluators determine a rating of attainment on a scale of 1 to 5, based on evidence obtained by examination of the project artifacts, or by observation of oral presentations. For each SO, the rubric provides a check-list of at least 7 rubric-points that may be earned. The number of

checked rubric-points, up to a maximum of 5, provides the rating. The evaluator may choose to record an *n/a* rating when there is insufficient evidence of a particular outcome in the artifacts. For SO (f) Teamwork and SO (h) Computing Tools, the ratings rely on rubrics completed by the students themselves (see section 4.1). For SO (g) Communication, the ratings are obtained by completing the *Oral Presentation Rubric*. For all remaining SOs, (a) through (e), the evaluator must examine the project artifacts. To assist the evaluator in locating the relevant sections, each project team completes an *Outcomes Check-List*<sup>11</sup> of entries that locate the available relevant artifact documentation.

#### RQ1 (*Applicability*):

Intuitively, the knowledge areas represented by SOs (b) CS Core, (c) Software Engineering, (d) Programming, (f) Teamwork, (g) Communication, and (h) Computing Tools are intrinsic to any team project with a presentation component. SOs (a) Foundations, and (e) Ethics, are less obviously essential. However, the Ethics component is listed among the CIS 4911 course outcomes, and the ABET SO (j) explicitly includes “*ability to apply mathematical foundations*” as part of the mix in the design of computer-based systems. Consequently, the *Senior Project Rubric* does enumerate examples of such features that might reasonably be expected in a capstone or life-scale project.

#### RQ2 (*Extendibility*):

The principal strategy employed in the *Senior Project Rubric* is enumeration of project features within each SO knowledge area. When qualitative assessment is desired, suitably designed subordinate rubrics yield metrics to be compared against pre-determined standards to yield check-marks. Thus, the *Senior Project Rubric* is two-tiered, and is extendible:

- By adding additional check-lists for new outcomes (1<sup>st</sup> Tier)
- By implementing subordinate rubrics to generate metrics to determine check-marks as described above (2<sup>nd</sup> Tier)

#### RQ3 (*Consistency*):

The rubric must allow consistent application with an expectation of identical results when applied conscientiously by different agents. Because the *Senior Project Rubric* has been applied to each completed Senior Project by at least two evaluators, the frequency of identical or near-identical comparisons may yield a significant measure of the rubric’s consistency.

## 5.2 Results

In the preceding section the *Senior Project Rubric* was characterized in terms of the following attributes:

RQ1 (*Applicability*):

Table 1 presents the distribution of Senior Project Rubric ratings over 22 completed projects, for each of the SOs. SO (a) is the only outcome to receive a rating of **n/a**, and only SOs (a) and (e) received any ratings of 1. The metrics for SO (a) and SO (e) are essentially counts of the number of instances of project features that lie within the domains of these outcome areas. Unless project specifications explicitly require application and documentation of features from the domains outside of the project focus-areas, students will not include them.

Student Outcome (SO)								
Rating	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
5	1	10	21	14	5	19	21	18
4	2	6	1	7	6	2	1	4
3	7	3	0	1	5	1	0	0
2	3	3	0	0	5	0	0	0
1	7	0	0	0	1	0	0	0
n/a	2	0	0	0	0	0	0	0

Table 1: Rating Frequency for each SO, All Data

The anecdotal evidence suggests that project clients have little interest in non-focus-area aspects. There is also evidence of this counter-productive tendency in the data for SO (b) CS Core. Incorporation and documentation of these aspects are not included unless project specifications, or mentors, require it.

As depicted in Table 1, the rubric provides good indicators of the utilization (whether high or low) of all SO areas. The rubric’s methodology is broadly applicable to obtain data on any SO whose subject area is utilized in the project implementation.

RQ2 (*Extendibility*):

The rubric is extendible in two ways. First, it can be extended to provide quantitative evaluation of a new SO by simply adding a 1<sup>st</sup> Tier check-list specific to the added SO. Second, it is may be extended to provide qualitative evaluation by assigning standards (minimum acceptability) for each bullet of an existing check-list, and designing a 2<sup>nd</sup> Tier rubric to provide qualitative ratings. These may be compared against the standards to provide check-marks for the high level 1<sup>st</sup> Tier assessment.

The single project presented in Summer 2011 was assessed using the Spring 2011 rubric, so the Summer 2011 data are counted with the Spring 2011 data. There were no rubric changes between Fall 2011 and Spring 2012.

The *Senior Project Rubric* was adjusted prior to application in Spring 2011, and again prior to Fall 2011. Table 2 shows the data for four semesters,  $\mu$  – mean and  $\sigma$  –standard deviation. Specifically:

- Spring 2011 – Addition of specific examples in check-mark criteria for both SO (a) Foundations, and SO (b) CS Core. This may account for improved statistics for SO (b) post Fall 2010.
- Fall 2011 - Incorporation of the 2<sup>nd</sup> Tier *Teamwork Peer Rating Rubric*, to seed the SO (f) Teamwork ratings. The change in the rating probably reflects a more realistic assessment in Fall 2010 and Spring 2011
- Fall 2011 - Incorporation of the 2<sup>nd</sup> Tier *Presentation Skills Rubric*, to seed the SO (g) Communication ratings. Informal feedback indicates altering the format for easier application.
- Fall 2011 - Incorporation of the 2<sup>nd</sup> Tier *Computing Tools Rubric*, to seed the SO (h) Computing Tools ratings. The rubric now more accurately reflects the breadth of students' experience with a variety of computing tools.

	Student Outcome (SO)							
	( a )	( b )	( c )	( d )	( e )	( f )	( g )	( h )
	<b>Fall 2010 (N = 7)</b>							
$\mu$	3.0	3.7	4.9	4.9	3.9	5.0	4.9	4.6
$\sigma$	0.96	1.25	0.17	0.17	1.37	0.00	0.17	0.44
	<b>Spring &amp; Summer 2011 (N = 6)</b>							
$\mu$	2.9	4.0	5.0	4.7	4.1	4.9	5.0	4.8
$\sigma$	1.48	1.15	0.00	0.47	0.73	0.19	0.00	0.37
	<b>Fall 2011 (N = 6)</b>							
$\mu$	1.1	4.5	5.0	4.7	2.5	4.8	5.0	5.0
$\sigma$	0.67	0.58	0.00	0.24	0.76	0.37	0.00	0.00
	<b>Spring 2012 (N = 3)</b>							
$\mu$	1.8	4.7	5.0	4.5	3.2	4.3	5.0	5.0
$\sigma$	0.24	0.47	0.00	0.71	0.62	0.94	0.00	0.00

Table 2: Comparison of Senior Project Data by Semester

RQ3 (*Consistency*):

This important attribute can be measured by employing paired-agent application. We define two consistency measures:

- $\delta=0$  – # of identical paired ratings
- $\delta=0,1$  – # of paired ratings that differ by 0 or by 1

The results, over all 22 applications, are summarized in Table 3:

	<b>FL 10</b>	<b>SP 11</b>	<b>FL 11</b>	<b>SP 12</b>	<b>ALL</b>
<b>#Pairs</b>	56	48	48	24	176
<b><math>\delta=0</math></b>	57.1%	72.9%	70.8%	75.0%	67.6%
<b><math>\delta=0,1</math></b>	87.5%	87.5%	93.8%	87.5%	89.2%

Table 3: Consistency Ratings by semester

$\delta=0,1$  is consistently just below 90%, but  $\delta=0$  has improved from an initial 57% to be consistently around the 70% mark. From Fall 2011 and after, the SO (h) ratings are entirely determined by student self-ratings, and SO (f) almost entirely by student peer ratings. Nonetheless, the semester  $\delta=0,1$  levels indicate that the rubric can be applied consistently.

Table 4 shows that at the outcome level, there is too high inconsistency for SO (a) and moderate inconsistency for SO (b) and for SO (e):

	Student Outcome (SO) (%)							
	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
$\delta=0$	27.3	50.0	90.9	59.1	45.5	90.9	86.4	86.4
$\delta=0,1$	68.2	77.3	100.0	95.5	81.8	95.5	95.5	100.0

Table 4: Consistency Ratings by Student Outcome

The semester trends for these outcomes are shown in Table 5:

$\delta=0,1$	FL_10	SP_11	FL_11	SP_12
# Pairs	7	6	6	3
SO (a)	85.7%	50%	83.3%	33.3%
SO (b)	57.1%	66.7%	100.0%	100.0%
SO (e)	100%	83%	66.7%	66.7%

Table 5: SOs (a), (b), (e):  $\delta=0,1$  consistency by semester

The trend for SO (b) is encouraging; the consistency seems to have responded positively to the rubric adjustments. However, the fluctuation in the consistency rating of SO (a), and the decline in consistency of SO (e) are not reassuring. The anecdotal evidence of evaluators is helpful. When not explicitly required, students fail to document SO (a), (b) and (e) aspects. A conscientious evaluator finds aspects that are not located by the students' check-lists; casual evaluators do not check for them.

## 6. Conclusion

We have described an approach to assessing attainment of Student Outcomes (SOs) of CS programs in capstone courses at Florida International University. Our approach is a two-tiered rubric. It allows a high-level 1<sup>st</sup> Tier enumerative methodology using check-lists that optionally may be supported by a 2<sup>nd</sup> Tier qualitative methodology. Our implementation of the *Senior Project Rubric* was initially single-tiered, but was soon supplemented by addition of a second tier to support assessment of three of the eight SOs.

This study has utilized the metrics produced by our rubric to demonstrate that the method is broadly applicable, and flexible to accommodate scaling and fine-tuning. The consistency of the metrics yielded by our instrument is often good. The exceptions underscore the prerequisite of purposeful mentorship that reinforces the capstone dimensions of the projects.

In the near term, the *Senior Project Rubric* will be modified to accommodate changes to our SOs. This study provides enough confidence in the applicability and scalability of the method, and consistency of its metrics, to suggest that retention of the instrument's methodology is practicable.

## References

- [1] ABET. Criteria for Accrediting Computing Program. [http://www.abet.org/uploadedFiles/Accreditation/Accreditation\\_Process/Accreditation\\_Documents/Archive/criteria-cac-2010-2011.pdf](http://www.abet.org/uploadedFiles/Accreditation/Accreditation_Process/Accreditation_Documents/Archive/criteria-cac-2010-2011.pdf) (Sept. 2012)
- [2] ACM/IEEE-Curriculum 2001 Task Force. Computing Curricula 2001, Computer Science. IEEE Computer Society Press and ACM Press, December 2001. <http://www.acm.org/education/curricula.html>
- [3] Al-Mubaid, H., Abeysekera, K., Kim, D., Perkins-Hall, S., Yue, K. 2011. A Model for Long Term Assessment of Computing and Information Systems Programs. *Information Systems Education Journal*, 9(3) pp 59-67. <http://isedj.org/2011-9/> ISSN: 1545-679X.
- [4] Abunawass, A., Lloyd, W., and Rudolph, E. 2004. COMPASS: a CS program assessment project. *SIGCSE Bull.* 36, 3 (June 2004), 127-131. DOI = <http://doi.acm.org/10.1145/1026487.1008031>
- [5] Ahmad, E., Raza, B., and Feldt, R.. 2011. Assessment and Support for Software Capstone Projects at the Undergraduate Level: A Survey and Rubrics, *Frontiers of Information Technology (FIT)*, 25-32, 19-21 Dec. 2011
- [6] Bailie, F., Marion, B., and Whitfield, D.. 2010. How rubrics that measure outcomes can complete the assessment loop. *J. Comput. Small Coll.* 25, 6 (June 2010), 15-25.
- [7] Clear, T., Goldweber, M., Young, F. H., Leidig, P. M., and Scott, K. 2001. Resources for instructors of capstone courses in computing. In *Working group reports from ITiCSE on Innovation and technology in computer science education (ITiCSE-WGR '01)*. ACM, New York, NY, USA, 93-113. <http://doi.acm.org/10.1145/572133.572135>
- [8] Farrell, A., Ravalli, G., Farrell, G., Kindler, P., and Hall, D. 2012. Capstone project: fair, just and accountable assessment. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education (ITiCSE '12)*. ACM, 168-173.
- [9] Richards, D. 2009. Designing Project-Based Courses with a Focus on Group Formation and Assessment. *Trans. Comput. Educ.* 9, 1, Article 2 (March 2009), 40 pages. <http://doi.acm.org/10.1145/1513593.1513595>
- [10] Sanders, K. E., and McCartney, R. 2003. Program assessment tools in computer science: a report from the trenches. In *Proceedings of the 34th SIGCSE technical symposium on Computer science education (SIGCSE '03)*. ACM, New York, NY, USA, 31-35. DOI = <http://doi.acm.org/10.1145/611892.611926>
- [11] Senior Project Rubrics and BS-CS links. <http://users.cis.fiu.edu/~pestaina/asee14.html> (Feb. 2014)
- [12] Southern Association of Colleges and Schools (SACS). <http://www.sacs.org/> (Sept. 2012)

## APPENDIX

### A. Senior Project Rubric for Outcome (a)

**Student Outcome (a):** *Demonstrate proficiency in the foundation areas of Computer Science including discrete structures, logic and the theory of algorithms*<sup>11</sup>

- \_\_\_ Project incorporates elements of mathematical reasoning or proof (Lemma, Theorem, Propositional Logic, First Order Logic, Mathematical Induction)
- \_\_\_ Project utilizes elements of discrete mathematics (Set Theory, Boolean Algebras, Combinatorics, Graph Theory)
- \_\_\_ Project utilizes some statistical procedure(s) to represent or summarize test data (Mean, Standard Deviation, Stem Plot/Histogram, Box Plot/Percentile-Graph)
- \_\_\_ Project utilizes some statistical measure(s) of system behavior or performance (Probability Distributions, Confidence Intervals, Hypothesis Testing)
- \_\_\_ Project design utilizes finite state diagrams to model system behavior
- \_\_\_ Project utilizes some aspect(s) of formal computer science (Automata, Turing Machines, Recursive Function Theory, Recursive Unsolvability)
- \_\_\_ Project utilizes some technique(s) of numerical analysis (Error Estimation, Interpolation, Numerical Calculus, Linear Systems, Matrix Algebra)

### B. Senior Project Rubric for Outcome (f)

**Student Outcome (f):** *Demonstrate the ability to work cooperatively in teams*<sup>11</sup>

To be completed by an evaluator:

- \_\_\_ Project presentation(s) included all team members equally
- \_\_\_ Project team activity is appropriately and adequately documented

To be completed from the data obtained from team members' peer evaluations:

Each team member rates each of the other members of their team individually on each criterion listed below on a scale of 1 to 5. The mean of all ratings for each criterion is recorded.

*The rubric item is checked only if the project (mean) score  $\geq 4.0$  for each of the 2 criteria.*

— Team members' roles were clearly defined and executed

<u>Criterion</u>	<u>Mean Score</u>
1: <i>Team members had clear understanding of expectations</i>	
2: <i>Team members maximized the use of their individual skill sets</i>	

— Project team set out and followed a schedule for timely completion

<u>Criterion</u>	<u>Mean Score</u>
3: <i>Team members complied with mechanisms to track progress</i>	
4: <i>Team members completed assignments in a timely fashion</i>	

— Project team negotiated consensus when needed

<u>Criterion</u>	<u>Mean Score</u>
5: <i>Team members showed respect for other team members opinions</i>	
6: <i>Team members were able to negotiate and compromise</i>	

— Project completion evidences equitable participation by team members

<u>Criterion</u>	<u>Mean Score</u>
7: <i>Team members contributed ideas and viewpoints</i>	
8: <i>Team members did their fair share of the work</i>	

— Team members shared responsibility for success and failure

<u>Criterion</u>	<u>Mean Score</u>
9: <i>Team members actively sought &amp; shared information from each other</i>	
10: <i>Team members were adaptable to changing requirements</i>	