ISSN: 2473-3857
v4 n4753

Principles of Program Revision for Computing Degrees: A Data-Driven 360 Degree Program Review Model

Svetlana Peltsverger
speltsve@kennesaw.edu

Lei Li
lli@kennesaw.edu

Rebecca Rutherfoord
brutherf@kennesaw.edu

Department of Information Technology
College of Computing and Software Engineering
Kennesaw State University
Marietta, GA 30060

Abstract

Computing degree programs need to be continually assessed and improved in order to keep current in the fast-changing computing field. Many of these changes are done on an ad-hoc basis without a structured assessment and improvement process. This paper presents an innovative program revision framework that examines every aspect of a degree program and uses an assessment process built on data analytics: a data-driven 360-Degree program review model. A data driven approach was used to collect both qualitative and quantitative data to evaluate an existing program and improve the program after implementing changes.

Keywords: Information Technology, Information Systems, program revision, curriculum development, data-driven 360 review.

1. INTRODUCTION

Developing an effective and sustainable revision process is critical for the continuous quality improvement of an educational program, especially for computing-related degrees. There are many factors motivating a degree program to periodically review and revise its curricula. First and foremost, an educational program has to comply with its accreditation requirements. A major accreditation agency for computing programs such as computer science, information technology, information systems and software engineering, is the Accreditation Board for Engineering and Technology (ABET) (www.abet.org). Most information systems programs are done through the Association to Advance Collegiate Schools of Business (AACSB) (www.aacsb.edu) accreditation. Both ABET and AACSB require their accredited programs to establish a mature assessment process which often trigger program revision to make sure that students achieve the learning outcomes set by the programs. Moreover, ABET accreditation mainly targets undergraduate programs. There is no accreditation currently available for graduate computing degrees. In this case, a well-established revision mechanism is even more important for the health of the degree program.

 ISSN: 2473-3857
v4 n4753

Computing is a dynamic and ever-changing field. Information technology (IT) and information system (IS) degree programs need to be constantly updated to keep up with latest development in industry and demand from the job market. For example, according to the Labor Insight platform by Burning Glass Technologies (http://burning-glass.com/), programming languages such as Python and R have gained more popularity in the job postings in the last three years. To make students more competitive in the job market, computing degree programs might consider adding those languages to their curriculum. And success of such an addition will largely depend on the program's revision process in place.

There is also fierce competition in the computing related education market. The online computing related degrees from for-profit institutions are booming. According to Burning Glass 3% of the 62,023 computing related baccalaureate degrees were conferred by the University of Phoenix-Arizona in 2015. The competition among non-profit institutions is also intense. There are many variations of computing degrees - computer science, information technology, information systems and software engineering, to name a few. A computing program needs to continuously improve the quality of its program in order to stand out from the competition which calls for an effective program revision process.

2. LITERATURE REVIEW

Different disciplines have different approaches to program revisions. For example, medical education uses a six-step revision approach (Thomas et al., 2015). Originally published in 1998, revised in 2009 and 2015, the six-step revision process includes 1) the study of demands for change to identify problems and general needs; 2) the use of surveys for targeted needs assessment; 3) setting goals and objectives based on competency based approach; 4) designing educational strategies; 5) identifying resources and the implementation; and 6) evaluation and feedback.

The Clemson Model of Industrial Engineering program revision starts with establishing goals for the revision based on the new program educational objectives. "The process then proceeds through a number of steps to integrate the discipline knowledge, group it into topics, make decisions about delivery, and use Bloom's Taxonomy to develop operational definitions within the course context of student outcomes". (Kimbler et al., 2006)

An Expert System Approach (Somkuwar, 2013) is based on finding gaps between the discipline module (competencies, skills and competency mapping) and competencies required by industries and jobs.
An Information Systems example of a curriculum development model (Noll et al., 2002) starts with a review of current available resources, followed by a review of the current environment, and needs/expectations of stake holders to identify critical skills and areas of knowledge to include into the revised curriculum.

The Curriculum 2010 initiative from Curtin University of Technology suggests that to have a 360-degree perspective of the course "health", an educator has to consider five voices (Oliver, 2008): 1) voice of current students; 2) voice of recent graduates; 3) voice of employers of recent graduates, and industry experts; 4) voice of competitors in the market; and 5) voice of current faculty/staff.

The model developed by Schleede and Lepisto (1984) uses four inputs: faculty philosophy & objectives, faculty resources, competitive analysis, and marketplace needs. Schleede and Lepisto's model has one final output, approved curriculum, where all problems and constraints are resolved. The model calls for periodic reviews and evaluation, but the model lacks implementation details and what should trigger a change process.

The literature also shows resistance to change and inertia by the program faculty across the disciplines. As Davis (Davis et al., 1998) points out, "A fundamental concept to initiate change in the curriculum revision process is to overcome resistance to change and the boundaries of self-interest. Curriculum change cannot occur without an "unfreezing" of faculty values and interests."

Although many papers on theoretical aspects of program review have been published, there are very few studies on the practical application and processes for adaptation. Most papers present examples of a particular degree program without proposing a process that others can follow to streamline the adaptation. In addition, while the data is often readily available given the proliferation of the information systems, data-drive decision making is rarely emphasized in the program review process.

Data analytics is becoming a future direction of many computing programs. By practicing what we preach, we can incorporate data analytics into the program revision process. Built on our

 ISSN: 2473-3857
v4 n4753

understanding of the exiting literature and our past assessment experience, we proposed a data-driven 360-degree program revision framework that is presented in this paper.

3. PROPOSED PROGRAM REVIEW MODEL

As illustrated in Figure one, the proposed 360 model has three layers that work together to effectively deliver a data-driven, systematic, comprehensive, and yet, customizable assessment process for revising and improving a computing degree program, such as IT or IS.

The *Program Layer* consists of all components of the program: admission standards, program mission, vision and outcomes, curriculum, student job placement, and technology infrastructure. The *People Layer* includes all constituencies involved: prospective and current students, alumni and recent graduates, Industry Advisory Board members, hiring managers, institutional support personnel, faculty, and staff. The *Data layer* includes all possible sources of internal and external data that can support or trigger a need for change.

A change in any of the components of the proposed model should trigger a review of all components of each layer. For example, in the case of a retirement of a key faculty expert, the department must evaluate how it will affect the program layer (e.g. course offering), the people layer (e.g. current students), and identify components of the data layer that can be used to track the effect of the change to prevent negative consequences for the program. A change in the data layer, for example, the trends of the job market should trigger review of the people layer (e.g. recent graduates, faculty expertise), and the program layer (e.g. curriculum recommendations).

The faculty are the center of the proposed review model. They initiate and manage the review process; interact with entities in the people layer, program and data layer; implement change, and follow up on the change(s) made. The four characteristics of the review model process are listed as follows:

1) A systematic review process. The proposed program evaluation consists of well-planned processes associated with a series of short-term and long-term goals. The overall goal of the review is the continuous improvement of the program. For ease of management, all program reviews should be aligned with the established review practices in the department. As shown in Figure two, there are two types of review processes: *predefined reviews* and *on-demand reviews.*

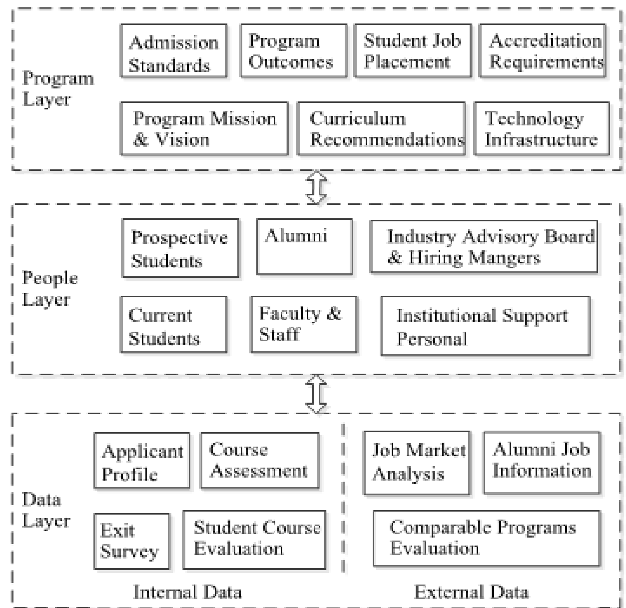

Figure 1. A Systematic and Data-driven 360 Degree Program Review Model

The *predefined reviews* can be further divided into end-of-the semester reviews and end-of-multi-year reviews. The program review can also be triggered by changes in any component of the proposed review model - such as assessment data indicates students have trouble with certain topics or, changes in the job market may require the program to move in a new direction. The event-triggered review is called an *on-demand review* and is handled by the department curriculum committee or ad-hoc committee appointed by the department chair. The findings discovered from the review process are stored in the knowledge repository, which is useful for future program reviews. Moreover, any type of review may result in changes to the program. It's important to close the loop by examining the effectiveness of the changes made in the next end-of-semester review period.

2) A 360-degree (comprehensive) review. An examination of all phases of the students' life cycle in the program is done, starting from students' admission to the program, their experience in the program, and their success after graduation. The different phases are interconnected and the review must be holistic. For example, the admission standards will have significant impact on a student's performance in

ISSN: 2473-3857
v4 n4753

the program. A student's GPA may correlate to his/her success in the job market. Feedback from alumni and existing students, on the other hand, will drive curriculum development or, changes in admission standards.

3) Data driven approach. A truly successful computing program cannot just rely on the model curricula of accreditation bodies and faculty good judgement. Instead, data is collected about the program and informed decisions are made based on the data analytics results whenever possible. Some examples of the internal data used includes applicants' profiles, student course assessment data, DFW and RPG rates, course evaluations, exit surveys, and feedback from Industrial Advisory Board (IAB) members. External data includes graduates' job information, analysis of the general job market, and evaluations of other comparable programs.

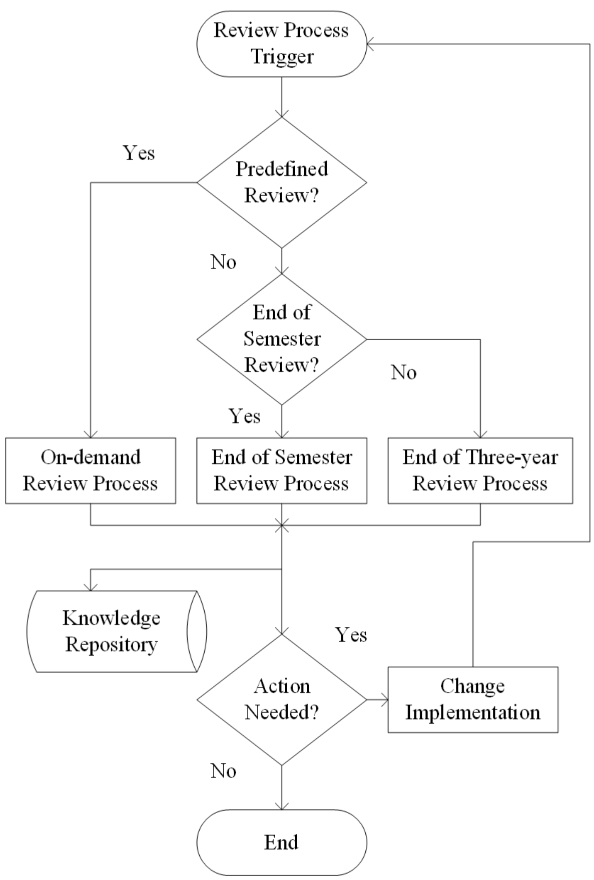

Figure 2. Review Process Flowchart

4) Result oriented. The action plan for change implementation should include timelines, measurable outcomes to assess the change and steps to achieve desired results.

The next section provides examples how the proposed 360 Data-driven model was successfully used in two revisions (2008 and 2017) of the Master of Science in Information Technology degree (MSIT) and, development of common first year introduction to programming courses for several undergraduate computing degrees.

4. EXEMPLARS OF THE REVIEW MODEL

4.1 Early MSIT Program Example

The MS in Information Technology program resides in the IT Department in the College of Computing & Software Engineering at a large public university in Georgia. Major changes since program creation in 1999 are summarized in Table one. The MSIT program is currently the largest master's degree at the university. One of the unique aspects of the MSIT degree is that a student can be conditionally admitted into the program without having a computing-related undergraduate degree. Before the first review in 2008, the program was a mix of CS and Management courses. The review process during 2007-2008 found that students, faculty and the IAB wanted primarily IT courses and not a mixture. The degree was changed using this data to require only IT courses with the caveat of taking up to three courses out of other approved master's programs. The program had 77 students in Fall 2008 and there were only 19 degrees conferred in FY2008. By 2012 both the enrollment and the number of the degrees increased by 40%. The changes made were positive for the growth of the program. However, this review was reactive in nature and did not use a structured review process.

4.2 2017-18 Revision of MSIT Using 360 Model

The second major change to the MSIT program began spring 2017. An extensive 360 Data-driven review was begun spring 2017 looking at elective courses students chose, comparator IT graduate degrees, Industrial Advisory Board recommendations for courses, faculty recommendations for courses, student demographics and a SWOT analysis completed by faculty. In addition, external data from Burning Glass helped define the types of jobs that IT students could find in the surrounding community and across the state after graduation. Data collected and analyzed showed that there were too many electives in the MSIT program which impacted faculty staffing and course offerings. In addition, the MSIT program required 36 hours for graduation, whereby data showed that many other IT graduate programs only required 30 hours.

4.2.1 Program Layer: The Admission Standard Review

The admission standard review is a predefined review process that is part of the three-year review. To start the process, a committee of graduate faculty discussed the need for revision of the existing admission standards and, agreed on the review procedures and timeline. In order to make informed decisions, the committee reviewed several layers of data analysis. The committee examined the alignment between the admission standards and program mission statement. Is there a mismatch? If so, how should the mission or admission standards be changed? The committee reviewed the admission standards from other comparable programs. To find comparable programs, our university has an approved list of Peer & Aspirational Comparator Universities.

<u>Connection to the Data Layer</u>: Is there anything we can use from other successful programs? It is easier now to find successful competitors. According to Burning Glass, in 2015 there were 12 universities in USA that conferred at least 100 master level degrees in Information Technology (CIP 11.0103) with the Carnegie Mellon University awarding 589 degrees, followed by University of Central Missouri with 235 degrees.

The analysis gave the committee a big picture of possible new admission standards. The data was drawn from the exiting knowledge repository or, collected on-demand. The correlations among students' admission profiles, their academic performance, and their job placement were examined. Some questions asked were: Will certain type of students be more likely to succeed in the program? What are the connections between students' academic performance and their success in the job market? To answer those questions, the review included students' admission data, including their undergraduate degree, GPA, their work experience, standard test scores, course assessment data and alumni job information. Following this portion of the review process in 2008, the department did not find a correlation between GMAT/GRE and students' success in the program, instead performance in foundation courses was the best predictor. Based on this analysis, the GMAT/GRE admission requirement was removed. In the following three years the department evaluated the results of this admission change and found no change in the success and retention rates of students. In the 2017 review admission standards were not changed.

4.2.2 Data Layer: Job Market Analysis

The goal of this analysis is to determine what technical skills are desired by the job market. For employment analytics and labor market information we used the Labor Insight platform by Burning Glass Technologies (http://burning-glass.com/). Many research questions can be used to analyze the job market. We provide two examples of labor trends questions for 2012-14 vs 2015-17.

First Question. Is the number of jobs requiring a master's degree for the occupations popular among MSIT graduates growing? The results for three positions, identified by tracking graduates using LinkedIn, computer and information systems manager (CIP 11-3021), computer systems analyst (CIP 15-1121) and information systems analyst (CIP 15-1122) are presented in Table two. The results show that in the last two years not only the number of job postings increased, but also, the ratio between the required education degree increased in favor of master degree holders.

Second Question: What are the most requested skills for master's degree holders? Table three shows that the most requested skill was, and is, the knowledge of software development principles. Several knowledge areas including JQuery and Big Data made a huge leap in ranking. Business knowledge including project management and business process & analysis stayed in the top 10.

<u>Connection to the People Layer</u>: To ensure that the MSIT program prepares students for the current job market and, that our program enhances career options for our graduates, we evaluated the performance of the program alumni. The department had created a LinkedIn group in 2012 and continues to track alumni success. In 2017, in the preparation for the degree revision, the department analyzed 600 LinkedIn alumni profiles and compared the career trajectory of our graduates (Davis & Rhodes & Baker, 1998)

<u>Connection to the Program Layer</u>: As the result of this review the program length was reduced to 30 credit hours to be compatible with other MSIT programs. The program now has two distinct options for students: the capstone option emphasizes practical aspects of IT, and a thesis option which requires a Research Methods course to strengthen research in IT for the thesis. In addition, seven outdated electives were replaced with three new electives: Practical Data Analytics, Research Seminar in IT, and IT Capstone.

Occupation and period	High school	Associate	Bachelor	Master % (#postings)	Doctoral
(CIP 11-3021) 2015-17	2.75	1.91	91.32	3.95% (56) 10%increase	0.07
(CIP 11-3021) 2012-14	2.25	1.80	91.07	4.60% (51)	0.27
(CIP 15-1121) 2015-17	7.86	3.71	85.51	2.43% (210) 76%increase	0.49
(CIP 15-1121) 2012-14	6.83	3.59	87.34	1.69% (119)	0.55
(CIP 15-1122) 2015-17	4.80	4.57	88.63	1.81% (102) 26%increase	0.19
(CIP 15-1122) 2012-14	8.26	5.31	83.65	2.62% = 81 postings	0.16

Table 2. Education (Minimum Advertised)

	Job Postings		Rank in	
Skill Clusters	2017	2012	12-14	15-17
IT: Software Development Principles	8,379	5,804	1	1
IT: SQL	6,282	4,459	4	2
IT: System Design and Implementation	6,022	5,304	2	3
Business: Project Management	5,192	4,817	3	4
IT: Java	4,751	3,539	8	5
Business: Business Process and Analysis	4,558	4,182	5	6
IT: Operating Systems	4,047	3,182	10	7
IT: Microsoft Office and Productivity Tools	4,043	3,828	6	8
IT: JavaScript and jQuery	3,704	2,228	16	9

Table 3. The most requested skill clusters for master's degree holders

In addition to studying our alumni career trajectories and success, we also presented suggested changes to the Industry Advisory Board. Their feedback was collected and vetted against data. For example, to check several requests from industry partners to produce more students who know legacy systems in order to replace their retiring baby-boomer mainframe personnel, the MSIT coordinator gathered local and state-wide data about the job market for those skills. For example, in the local Labor market Area the number of job postings requiring COBOL declined almost 50% between 2015 and the last 12 months (Burning Glass). The data clearly showed that there was a miniscule need for these skills. Thus, the certificate for mainframe computing was not created.

In the last two years, the department hired several faculties with new research and teaching interests. The hiring was aligned with industry needs, and then the program changes followed.

4.2.3 People Layer: IT Foundation Courses Review

Almost half of the MSIT program students are non-conventional students who don't have computing backgrounds and want to change their career to IT fields. Such an admission approach not only responds to the NSF's call for universities to offer STEM bridge programs, but also is the key success factor for the MSIT program. The career changing students are required to take up to four foundation courses. To better serve this special student group, the department revised the number and the content of the foundation courses at least three times between 2008 and 2017.

Since the revision in 2008, the department has closely tracked student performance in foundation courses and their success in the consecutive courses. Student course outcomes were tracked and, the foundation courses were improved to ensure that students completing those courses satisfy some major curricular components of the IT baccalaureate level program. One of the implemented changes is expansion of coverage of web development that includes integration. Another change removed the Information Security foundation course, and included its outcomes into all other foundation courses.

	Sample size	Avg. GPA
Conventional student	27	3.47 (0.33)
Non-conventional student	46	3.74 (0.17)

Note: the number in the parenthesis is the variance of the average.

Table 4. Academic Performance of Conventional Students vs. No-conventional Students.

 ISSN: 2473-3857
v4 n4753

Connection to the Data Layer: Following our data driven approach, qualitative and quantitative data were collected to evaluate the changes in foundation courses. End of semester course assessment reports are used to track students' performance in individual courses. Table four compares the accumulative GPA of non-conventional students with the ones of conventional students who have a CS/IT/IS background. The T-test (assuming unequal variance) shows that non-conventional students' average GPA is statistically higher than the ones of conventional students. We also used student course surveys and exit surveys to collect qualitative data relate to the foundation courses.

Connection to the Program Layer: both quantitative and qualitative data showed that the foundation courses serve the needs of career-changing students. The data also confirmed that we should continue our approach admitting non-conventional students to the MSIT program.

4.2.4 Results of using the 360 Model

Data collected showed that there were too many electives in the MSIT program that impacted faculty staffing and course offerings. In addition, the MSIT program required 36 hours for graduation, whereby data showed that many IT graduate programs only required 30 hours. Data also showed that many elective courses were either out-of-date, or no longer important for an IT graduate program. With the extensive data that was collected, the faculty revised the MSIT program to require 30 hours, and dropped the total number of elective courses from 24 to 20. Seven courses were removed, and three new courses were added to meet the needs of industry and students. Students now choose either a Capstone Option, or the Thesis Option. Students may only take one course outside of IT from approved majors. This major curriculum change will take effect fall 2018. A summary of MSIT revision history is shown in table five.

4.3 First Year Programming Course Example

Another example of using the 360 model took place during spring, summer and fall 2017. The college had anecdotal data that many of our students were not successful in passing the first two programming courses in all of our computing majors. The college decided to take the formalized approach to look at ways to improve this problem.

Data Layer: First, statistics and reports were generated during spring 2017 using the student record system to get actual numbers of students who passed CS1 and CS2 with a grade of "C" or better. Additional statistics were gathered to see how students who passed with a grade of "C" or "B" or "A" did in successive courses. The data showed several things:
(1) The grades of "D", "F" or "W" in CS1 were around the 50% rate. In other words, half of the class was not passing.
(2) The students who received a "B" or higher in CS 1 and CS 2 had a higher percentage of doing well in subsequent courses.
(3) The students who received a "C" in CS1 and/or CS2 had a much lower percentage of doing well in subsequent courses.

Courses	Pre 2008	2008 Revision	2017 Revision
Foundation courses	7 (CS, IT, MGNT)	3 (IT) increased to 4 in 2012	4 (IT)
# of courses	12	12	10
Required courses	7 (MGNT, SWE,IT)	5 (IT)	4 (IT)
Required technical elective	0	1	0
Program options	none	thesis or non-thesis	thesis or capstone
Non-IT courses	Up to 5	3	1

Table 5: MSIT Revision History

People Layer: Using this and additional data, the college created a volunteer committee (with representatives from each major) to come up with possible solutions to this problem. The committee met late spring, summer and early fall and came up with several actions to address the problem.

Program Layer: The solutions for first year courses included:
1. Each computing program would now have a "gate" in order to choose a computing major during the first year. Four programs chose to allow students into the chosen major if they received a "B" or better in both CS 1 and CS 2. One program opted to gate their program with a "B" or better in CS 1. (admissions)
2. A new elective "Intro to Computing" was added to the curriculum and through orientation and advising, students are encouraged to take the course. Research showed a correlation in taking

this type of course and success in CS 1 (Brown, 2013) (curriculum requirements).
3. In addition, the college hired four new faculty to only teach the Intro to Computing, CS 1 and CS 2 courses. (this ties to the People layer-faculty)
4. Graduate assistants were hired to assist the first year teaching faculty as student mentors, tutors, lab assistants and graders. (this ties to the People layer – support) The new system goes into place fall of 2018, so results of the actions will not be evaluated until 2019.

5. ADAPTATION RECOMMENDATIONS

As demonstrated in section four, conducting a comprehensive and data-driven program review is a complex, and time-consuming process which requires careful planning and commitment from the faculty and program administrators. Below is list of recommendations on how to successfully adopt the proposed review model.

A. Build a review process that best fits your program. An academic program often has its unique set of characteristics. Design the proposed program review framework to be flexible and customizable as the framework defines "what needs to be done", not "how to do it". Programs that are interested in adopting the 360 Data-driven proposed model can decide on review activities and implementation methods to fit their environment. For example, if a program already has a well-established alumni database, it might not necessary to go to LinkedIn to collect alumni's career information. The best review model for a program is a well-planned one based on its own situation.

B. Focus on the essential activities. One danger of doing data-driven comprehensive review is getting lost in the sea of data and review activities. It's important to focus on the essentials. The core of any academic program assessment is that students achieve the learning outcomes set by the program. So, systematic assessments of learning outcomes at course level and program level is essential and, all other assessment activities are built on it. Another key in program review is to make sure there is a process to "close the loop". For an issue identified in a program review activity, there is should be a built-in mechanism to evaluate and implement, if applicable.

C. Be efficient with the data. One of the biggest concerns in adopting the proposed review is the data. It is labor intensive and time consuming to collect, process, and analyze data. While there is no silver bullet for this matter, we have several suggestions: 1) having data collection built in when creating an assessment plan. For example, when your review plan includes course level assessment, a unified form should be designed in a way the assessment data can be easily entered by the faculty and easily aggregated later on at the program level; 2) data collection is distributed throughout the program assessment cycle and not just concentrated in small periods of time. This is important not only for the success of the review plan, but also for increasing faculty buy-in in the assessment activities; 3) automate the data collection if possible. For example, exit surveys can be done through online survey software. You can develop a web crawler to automatically collect alumni's career information from LinkedIn. Doing this requires significant time investment at the beginning. However, the data collection will be easier later on if an automated process is in place; 4) analyzing data requires good tools. Some are free and other tools, such as Burning Glass, require hefty license fees. A program needs to consider when the tools will be used in the planning phase to minimize the cost.

D. Be prepared for resistance and limitations. There might be resistance to change from some faculty from claiming shortage of time who are not willing to change course content or their teaching style. It's often easy to underestimate the need for resources, such as, not having enough qualified faculty to teach or funds for equipment in highly specialized courses.

In summary: develop a customized assessment process to fit a program with focus on the essential activities – more focused assessment activities are better than less; use data-driven decision making in the assessment.

6. CONCLUSION

A structured review is important for continuous improvement of an academic program. This is particularly true for computing programs given the dynamic nature of the field. The fact that ABET and IEEE/ACM have yet to publish any criteria for graduate computing programs is an additional reason to use a structured process. In this paper, we discussed a comprehensive and data-driven model for the systematical review of IT programs. We demonstrated the effectiveness of our approach using examples. We plan to continuously improve and validate the revision model based on further research and the data analysis. We believe that the 360 data-driven model will continue to help us improve the quality of our programs and, be useful to other computing programs and the academic community.

7. REFERENCES

Brown M., (2013). CS0 as an indicator of student risk for failure to complete a degree in computing. J. Comput. Sci. Coll. 28, 5 (May 2013), 9-16.

Davis, D. C., Rhodes, R., & Baker, A. S. (1998). Curriculum revision: reaching faculty consensus through the nominal group technique. Journal of Nursing Education, 37(7), 326-328.

Kimbler, D. L., Gramopadhye, A. K., Kurz, M. E., Melloy, B. J., & Leonard, M. S. (2006), Implementing Curriculum Renewal from Courses through Curriculum Revision, IIE Annual Conference Proceedings. Institute of Industrial and Systems Engineers (IISE).

Noll, C.L. & Wilkins, M. (2002). Critical Skills of IS Professionals: A Model for Curriculum Development. Journal of Information Technology Education. 1, 3 (2002).

Oliver, B. (2008). 360-degree Feedback on Courses : Needs Analysis for Comprehensive Course Review. Proceedings of the Australian Universities Quality Forum 2008 (2008), 55–60.

Thomas P. A., Kern, D. E., Hughes, M. T., Chen B. Y. (2015). Curriculum Development for Medical Education a Six-Step Approach. The John Hopkins University Press. (2015).Schleede Jr, J. M., & Lepisto, L. R. (1984). Marketing curriculum development: Model and application. Journal of Marketing Education, 6(1), 2-9.

Somkuwar, V. (2013), Curriculum Design for Technical Institutions -An Expert System Approach. International Journal of Engineering Sciences & Research Technology. 2, 2 (2013), 281–285.