International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor (2018): 7.426

An Insight on the Evaluation of the Programmer Applicants: A Review

Fayiz Momani[1], A.A. Zaidan[2]

[1, 2]Universiti Pendidikan Sultan Idris, Tanjung Malim 35900, Malaysia

Abstract: ***Selecting the best software programmer applicant is a challenging task because the number of graduates from information technology department has increased rapidly due to the advancement in information and communication technology (ICT). This increase has resulted in severe competition between programmer applicants for jobs. This article is to review the studies relating to computer science and computer engineering evaluation applicants. Moreover, it describes the classification of applicants based on the software development life cycle phases. Programmer is the main axis of the study into the implementation phase.***

Keywords: Programmer, computer science, computer engineering

1. Introduction

The number of graduates from information technology department has increased rapidly due to the advancement in information and communication technology (ICT) (Al-Masri & Ahmad, 2017). This increase has resulted in severe competition between newly graduates for jobs. Accordingly, companies that are in need for employees has no specific system to select amongst the candidates. When the company chooses the best applicant, the performance of company will be increased. This study aims to review the previous articles that is related to computer science applicant's evaluation and computer engineering applicant's evaluation and the focal point of this study is the programmer applicant.

The Implementation phase is the most technical phase in the software development (Pirzadeh, 2010), and it addresses human related issues that are linked to the Implementation phase in terms of the technical aspect. This phase comprises the conversion of software design into a set of programs or program units. As stated by (Deliktas & Ustun, 2015; Pirzadeh, 2010), among the examples are programming language, databases, functions, coding, pair programming, personality, productivity, dependability, reuse, end-user programming and methods. For SWEBOK, three key subcategories for Implementation are considered: Constructing for verification, Coding, and Reuse (Pirzadeh, 2010). Constructing for verification has more linkage with the early concerns of the implementation including programming language choice, platforms and tools. Coding on the other hand, is the primary portion in the phase of implementation in which the concerns of developers are linguistics and visual issues. In reuse, developers perform unit and integration testing to ascertain the utilization of the software by mixing it to their system. These categories have been utilized in the form of extraction (Pirzadeh, 2010).

Computer programs are created by computer programmers, and as stated by (Paudel, 2016), the construction of almost all programs necessitates the use of programming language like C++, Python as well as Java. As described by (Paudel, 2016), the aforesaid programming languages are akin to the language used in day-to-day speech and writing. In particular, a compiler program is used to translate the instructions from user into binary code (machine code). This binary code is understood by the computer as zeroes and ones and the computer will perform what is necessitated.

Programming comprises the situation description process, and the description is refactored based on a set of computational formalisms (Blackwell, 2002). A human-centered perspective defines the notion of programming as the process of converting "a mental plan into one that is compatible with the computer" (Myers et al., 2016).Worded differently, programming comprises the coding of plan into a language that can be read by computer. The plan is then translated by the computer into program whose functions are comprehensible and usable by user according to what is seen by user on the screen. The programmer is responsible to translate the mental plan into computer program.

A programmer, computer programmer, developer, coder, or software engineer writes computer software or to a generalist who produces code for numerous software types. Meanwhile, a programmer analyst practices or professes a formal approach to programming. Further, the main computer language that programmers use includes Assembly, COBOL, C, C++, C#, Java and Python. As explained by (Paudel, 2016), a Web developer, a software developer, mobile applications developer, software engineer, embedded firmware developer, computer scientist, or software analyst, is also a programmer.

This article is a review of the studies that evaluate the computer science and computer engineering applicants through specifying the application output of each study. In addition, general gap is determined by method type and specific issues is determined by the technique used in each study.

2. Literature Review

This section presents studies related to the method used in evaluating the programmer applicants. (Molins-Ruano, Rodriguez, Atrio, & Sacha, 2016) confirmed that when evaluated based on computer science skills, the results provide additional applicable information when compared with a group of experts. The additional information obtained by this model allows the identification of users that do not have the suitable formal knowledge even when they are capable of good scores according to old-style evaluation systems.

To use the method, the authors established an adaptive test that gives additional information about the users' behaviour and formal knowledge. As well as, the author uses e-valUAM method to evaluate the applicant in Expert behaviour instead of the traditional scores based only on the number of correct answers. (Sharma, Banerjee, Mandal, & Vikas, 2015) proposed an innovation to construct electronic think banks to avoid evaluation instrument specific and estimation errors and enhance the quality of appraisals. The authors proposed 12 remarkable item that are consistent with the programmed assessment. The method of assessing the understudy's reaction consequently is examined in detail.

The automated assessment is a good approach to achieve legitimacy and dependability for extensive scale appraisals, by using the survey in automated assessment application. (Felemban, Gardner, & Callaghan, 2016) combined a computational tool that mixes software agents and natural agents (users) with an ontology line that supports the identification of learning evidence from collaborative activities that mimic classroom observation by proposing a method for evaluating student learning within collaborative groups in 3D virtual worlds (VWs) using four methods ; Multi-agent and Ontology-based Approaches, Multi-agent Systems, Natural Agent (NA) and Ontology Agent (OA) in 3D virtual environments application. (Kakani, Dalal, & Dabhi, 2016) explained the existing assessment methods and offered a solution by considering the confidence and satisfaction levels of assessors with the use of fuzzy logic. (Baneres, 2016) presented a contextual investigation of an expectation framework regarding a first-year computer science course about concepts of digital system fundamentals. The point of the methodology is to encourage students and educators amidst the instructional procedure with the final goal of effectively passing the course, using predictive models' method in evaluating. (Gurupur, Jain, & Rudraraju, 2015) proposed an instrument that can be used to adequately assess student learning results via idea maps and Markov chain analysis. The principal motivation behind this instrument is to propel the utilisation of artificial intelligence methods by using idea maps and Markov chains in assessing a student comprehension of a specific point of study, using Concept maps and Markov chains method in evaluating. (Sharma et al., 2015) proposed a suite of benchmark PC data innovation assignments to assess student execution in early data innovation courses. Some future difficulties involved in expanding this suite to address advanced programming courses are examined. (Cain, 2013) This work investigated the improvement of evaluation criteria for an early programming unit that utilised portfolio appraisal to execute productive arrangement. After initial errors, powerful evaluation criteria that empowered quick and exact appraisal of student portfolios were identified in Portfolio assessment application. (Abdeljaber & Ahmad, 2017) proposed an assessment method and used it to evaluate student performance through exams and quizzes. They recommended improvements for the overall assessment process. Their finding shows the efficiency and flexibility of the proposed assessment method, using framework method in evaluating. (Damaj & Yousafzai, 2016) proposed a framework for the outcome of undergraduate computer engineering student assessments based on design experience. However, they discussed the skills, experiences and abilities for senior design in a bachelor of engineering program, using framework method in evaluating. (Yildiz & Baba, 2014) developed a new model based on the fuzzy multi-criteria method, and they mentioned that fuzzy enables more reliable decision making and evaluation. This model evaluates students in laboratory activities. However, they used personal, group and peer assessment methods to confirm that the fuzzy method could deliver a better evaluation system than classical systems, using Fuzzy 'refinement process' method for evaluating in Laboratory applications. (Macek & Kom, 2012) used software engineering students as a sample. They discussed numerous techniques used to evaluate teamwork among the students, using Four techniques method for evaluating in Real company. These studies provided different applications based on their evaluation criteria. However, considerable work is still not providing any framework that can select the best software programmer applicant based on multi-evaluation criteria analysis (Zughoul et al., 2018).

3. Findings

Table 1.1 shows the studies that are relating to the computer science and engineering evaluation applicants. It involves three main points; the techniques used in the evaluation or the selection of the programmer applicant, the output model of evaluation that specifies the general gap of the review, and the application utilized in the study.

Reference	Method used	Method		Application
		Evaluation	Selection	
(Molins-Ruano et al., 2016)	e-valUAM	✓		1-Experts' behaviour instead of the traditional scores based just on the number of correct answers 2-The test is applied by the e-valuam application
(Banerjee, Ramanathan, & Rao, 2015)	Survey	✓		Automated assessment

(Felemban et al., 2016)	1) Multi-agent and Ontology based Approaches 2) Multi-Agent Systems 3) Natural Agents (NA) 4) Ontology Agent (OA)	✓		3D virtual environments
(Kakani et al., 2016)	Fuzzy logic techniques	✓		
(Baneres, 2016)	Predictive models	✓		Predictive models
(Gurupur et al., 2015)	Concept maps and Markov chains	✓		Concept maps and Markov chains
(Sharma et al., 2015)		✓		
(Cain, 2013)		✓		Portfolio assessment
(Abdeljaber & Ahmad, 2017)	Framework	✓		Computer science (CS) programme of a Saudi university was taken as a case study. The programme offers 31 major courses in CS, and we implement the proposed assessment method for all courses.
(Damaj & Yousafzai, 2016)	Framework	✓		
(Yildiz & Baba, 2014)	Fuzzy 'refinement process'	✓		Laboratory applications
(Macek & Kom, 2012)	Four techniques	✓		Real company

The critical review in Table 1.1 above shows that (12) studies are related to computer major and are grouped into two domains namely; (9) of these studies are based on the computer science domain. While, (3) of them are based on the computer engineering domain. The entire studies provided different applications based on their evaluation criteria.

4. Conclusion

This study reviewed twelve articles related to computer science and computer engineering domain, from these articles the study found that each article has different methods used on evaluation, and all papers using only evaluation in the method. Therefore, the papers have been reviewed are applied in different applications and domains. Therefore, a framework for selecting the best programmer applicant has not yet implemented.

References

[1] Abdeljaber, H. A., & Ahmad, S. (2017). Program Outcomes Assessment Method for Multi-Academic Accreditation Bodies: Computer Science Program as a Case Study. *International Journal of Emerging Technologies in Learning, 12*(5).

[2] Al-Masri, O. H., & Ahmad, M. S. (2017). A Conceptual Framework for a Problem Resoulution Support System (PReSS). *Journal of Advances in Information Technology, 8(2), 148-153.*

[3] Baneres, D. (2016). *Towards a particular prediction system to evaluate student's success.* Paper presented at the International Conference on P2P, Parallel, Grid, Cloud and Internet Computing.

[4] Banerjee, S., Ramanathan, C., & Rao, N. (2015). *An approach to automatic evaluation of higher cognitive levels assessment items.* Paper presented at the MOOCs, Innovation and Technology in Education (MITE), 2015 IEEE 3rd International Conference on.

[5] Blackwell, A. F. (2002). *First steps in programming: A rationale for attention investment models.* Paper presented at the Human Centric Computing Languages and Environments, 2002. Proceedings. IEEE 2002 Symposia on.

[6] Cain, A. (2013). *Developing assessment criteria for portfolio assessed introductory programming.* Paper presented at the Teaching, Assessment and Learning for Engineering (TALE), 2013 IEEE International Conference on.

[7] Damaj, I., & Yousafzai, J. (2016). *Simple and accurate student outcomes assessment: A unified approach using senior computer engineering design experiences.* Paper presented at the Global Engineering Education Conference (EDUCON), 2016 IEEE.

[8] Deliktas, D., & Ustun, O. (2015). Student selection and assignment methodology based on fuzzy MULTIMOORA and multichoice goal programming. *International Transactions in Operational Research*.

[9] Felemban, S., Gardner, M., & Callaghan, V. (2016). *An event detection approach for identifying learning evidence in collaborative virtual environments.* Paper presented at the Computer Science and Electronic Engineering (CEEC), 2016 8th.

[10] Gurupur, V. P., Jain, G. P., & Rudraraju, R. (2015). Evaluating student learning using concept maps and Markov chains. *Expert systems with applications, 42*(7), 3306-3314.

[11] Kakani, B., Dalal, D., & Dabhi, A. (2016). *Improved solution on students answer sheet assessment using fuzzy rules.* Paper presented at the Advances in Signal Processing (CASP), Conference on.

[12] Macek, O., & Kom, M. (2012). *Evaluation of student teamwork.* Paper presented at the Software Engineering Education and Training (CSEE&T), 2012 IEEE 25th Conference on.

[13] Molins-Ruano, P., Rodriguez, P., Atrio, S., & Sacha, G. (2016). Modelling experts' behavior with e-valUAM to measure computer science skills. *Computers in Human Behavior, 61*, 378-385.

[14] Myers, B. A., Stefik, A., Hanenberg, S., Kaijanaho, A.-J., Burnett, M., Turbak, F., & Wadler, P. (2016). *Usability of Programming Languages: Special Interest Group (SIG) Meeting at CHI 2016.* Paper presented at

the Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems.

[15] Paudel, S. (2016). VULNERABLE WEB APPLICATIONS AND HOW TO AUDIT THEM: Use of OWASP Zed Attack Proxy effectively to find the vulnerabilities of web applications.

[16] Pirzadeh, L. (2010). Human factors in software development: a systematic literature review.

[17] Sharma, K., Banerjee, K., Mandal, C., & Vikas, I. (2015). *A benchmark programming assignment suite for quantitative analysis of student performance in early programming courses.* Paper presented at the MOOCs, Innovation and Technology in Education (MITE), 2015 IEEE 3rd International Conference on.

[18] Yildiz, Z., & Baba, A. F. (2014). *Evaluation of student performance in laboratory applications using fuzzy decision support system model.* Paper presented at the Global Engineering Education Conference (EDUCON), 2014 IEEE.

[19] Zughoul, O., Momani, F., Almasri, O. H., Zaidan, A. A., Zaiden, B. B. Alsalem, M. A., Albahri, O. S., Albahri, A. S., Hashim, M. (2018). Comprehensive Insights into the Criteria of Student Performance in Various Educational Domains. *IEEE Access, 6, 73245-73264.*

Author Profile

Momani, F. M. S. M received his B.Sc. degree in Computer Information System in 2011 from Philadelphia University, Amman, Jordan. Then, he received his M.Sc. degree in Information Technology in 2014 from Universiti Tenaga Nasional, Bangi, Malaysia. Currently, he is candidate Doctor of philosophy (PHD) student in Universiti Pendidikan Sultan Idris (UPSI), Tanjung Malim, Malaysia. His research areas are: Student evaluation, student assessment, student appraisal, educational domains.

A.A. Zaidan received his first-class B.Eng. degree in Computer Engineering in 2004 from University of Technology, Baghdad, Iraq. Then, he received his M.Sc. degree on Data Communications and computer network in 2009 from University of Malaya, Malaysia. Then, following his Ph.D. degree on artificial intelligence in 2013 from Multimedia University, Malaysia. Currently, he is in working as senior lecturer at Department of computing, University Pendidikan Sultan Idris. He led or member for many funded research projects and He has published more than 130 papers at various international conferences and journals. His research areas are: Student evaluation, student assessment, student appraisal, educational domains.