UDC: 37-042.4:004.041
DOI: 10.15587/2519-4984.2024.304973

FOREIGN EXPERIENCE OF SOFTWARE TESTING ENGINEERS TRAINING: ANALYSIS OF EDUCATIONAL STANDARDS AND PROGRAMS OF AMERICAN UNIVERSITIES

Oleksandr Gura

The article aims to explore the U.S. educational system's approach to training software testing specialists, highlighting specific features and methodologies. The study involves analyzing key standards for IT engineer training in the USA, such as the ACM/IEEE Computer Science Curricula, ABET Accreditation Criteria, ISTE Standards for Educators, and CSTA K-12 Computer Science Standards, their overall context and targeting, as well as the inclusion of both fundamental and IT-specific disciplines into their content. In the study of each standard, particular emphasis was placed on defining the place and teaching methods of theory and practice of software testing in the university curriculum guidelines. As it was revealed in the analysis, despite the extremely similar interpretation of basic concepts and practices in the context of software testing, opinions about the methodology of their teaching and the need to study as a separate discipline, differ in a set of reviewed standards.

Additionally, the research examines training programs at leading American universities, including MIT, Stanford, and Carnegie Mellon University, to identify how software testing and quality assurance are reflected in their curricula, covering full-time training programs for students in computer science, software engineering and similar related fields. The analysis examined programs in both general and basic disciplines in the context of information technology, and specific to a certain technological or organizational area, such as mobile device development or IT project management. The content of the subjects, the list of modules and lectures, as well as tasks for extracurricular work, available in open sources, was also analyzed and searched for topics related to quality assurance or software testing. As revealed in the study, in most technical universities testing is not covered within a separate discipline, but is taught as a part of ones related to development practices. However, some training programs, in particular the Master's level, still have disciplines, devoted exclusively to quality control, testing management and automation

Keywords: *software engineering, software testing, educational standard, training program, foreign experience*

How to cite:
Gura, O. (2024). Foreign experience of software testing engineers training: analysis of educational standards and programs of American universities. ScienceRise: Pedagogical Education, 2 (59), 73–78. doi: http://doi.org/10.15587/2519-4984.2024.304973

1. Introduction

The rapid expansion and rising demand for information technologies across various areas, including both business and education, is a phenomenon being observed for the latest years not only on relatively young Ukrainian market, but all across the world. Europe and America, being the areas with leading global IT enterprises, significantly influence the trajectory of informational infrastructure development, software engineering methodologies, and newly appearing trends and frameworks, including ones related to education: the maturation and rapid growth of the IT market inherently drive the need for skilled professionals, pushing the universities to introduce new specialties, integrate additional disciplines,and update curricula to align with contemporary market demands and supply competent personnel.

In Ukraine, the IT industry is relatively new and heavily influenced by international trends: the majority of Ukrainian IT companies operate as branches or business partners of European and American global corporations, taking over operational frameworks, technological choices, team dynamics, and software development methodologies. This influence extends to the domain of software testing, which is regarded as one of essential competenciesin software development sphere, and, accordingly, one of the most popular and in-demand professions in the Ukrainian labor market in the field of information technology.

Given the dynamic nature and inclination of the IT industry to constant changes and the continuous adoption of new approaches, tools and practices, the educational sphere is forced to constantly focus on business to keep its training programs relative to modern realities. As an ambassador in the business segment of information technology, the United States of America is also shaping trends in the context of training new professionals, integrating ever-new practices, approaches, and curricula into its IT education system. The orientation of modern Ukrainian IT-sphere to Western requirements, as far as the same legal status of the profession of software engineer in general and software tester in particular, allow to conclude similar requirements to the process of training of specialists in the sphere, which, accordingly, determines the relevance of studying the American experience of training software testing engineers.

2. Literature review

The context of foreign pedagogical experience has been relevant in the Ukrainian scientific space through all the history of the formation of the domestic education system. The study and adaptation of the most effective educational practices and concepts is observed primarily in the general context of the problems of contemporary pedagogy and training of young professionals, such as the assessment of the quality of education or its individual national aspects [1, 2]. Special attention is also given to the field of information technology, both the general history of the growth of the industry, and its integration with higher education [3], and the analysis of specific practices and methodologies for the training of future software engineers [4]. The problem of software testing is much less popular among modern scientists, which is caused, in the first place, by the not fully defined status of the sphere of quality assurance itself, considered by many as an appendix to the broader competence of the software development engineer. It is worth noting that this thesis is emphasized primarily in the work of American colleagues, raising in their scientific work the question of the status of the sphere of quality assurance and integration of its theoretical and practical aspects in the training programs of American universities. [5–7].

Considering a rapid growth in demand and popularity of the profession of software testing engineer in the domestic IT market, and, accordingly, the growing demand for quality training of relevant specialists, the Ukrainian educational system needs to constantly adapt the content of its educational programs to the new realities of the market. In our opinion, analysis of foreign, particularly American, experience of training software testing engineers can be a valuable and significant part in this process, which determines the relevance of the study.

3. The aim and objectives of the research

The purpose of the article is to study the experience of the educational system of the United States of America in training software testing specialists and highlight its specific features.

To achieve this goal, the following tasks were identified:

1. Define and study the key standards for training IT engineers in USA. Determine the place of software testing in the content of these standards.

2. Identify key representatives of American higher education institutions in the field of information technology. Analyze training programs in the specialties “Computer Science” and “Software Engineering” for the inclusion of modules or even individual disciplines, devoted to the area of software testing and quality assurance.

4. Materials and methods

To achieve the goals defined above, various sources and methodologies were used, including the analysis of scientific literature, educational standards, and educational curriculums of the universities under the research. Various resources were used to find a data for the research: search engines (e.g., Google, Bing), scientific databases (e.g., Google Scholar), digital repositories of higher educational institutions (MIT, Stanford etc.). Artificial Intelligence technologies, including Chat GPT 3.5, Quill Bot AI Translator and Quill Bot Grammar Checker, were used for more precise data aggregation and utility. During the research, various theoretical methods were considered: abstraction and concretization, analysis and synthesis, comparison, structuring, induction, and deduction.

5. Research results and discussion

In the United States, several fundamental standards define the requirements for preparing new specialists in the software development area. The coexistence of multiple educational standards, such as the ACM/IEEE Computer Science Curricula, ABET Accreditation Criteria, ISTE Standards for Educators, and CSTA K-12 Computer Science Standards, reflects the diverse and multifaceted nature of education in the United States. Each set of standards serves a specific purpose and audience, addressing different aspects of the educational continuum. Legally, these standards are not mandated by federal law but are highly influential in shaping educational practices and policies. Universities and educational institutions adopt these standards to align with best practices, meet accreditation requirements, and ensure that their programs are competitive and comprehensive.

ACM/IEEE Computer Science Curricula – a standard, which outlines guidelines for undergraduate and graduate programs in computer science and related fields, covering core knowledge areas, learning outcomes, and curricular recommendations. The first version was published in 1968, and it has since undergone multiple revisions to reflect new approaches in technology and changes in the computing field. The latest version includes ongoing updates and revisions to ensure relevance and alignment with industry trends, being systematically overseen by ACM and IEEE committees. The ACM/IEEE Computer Science Curricula outline essential knowledge areas, such as algorithms, programming languages, software engineering, databases, and computer architecture. It emphasizes hands-on learning, problem-solving skills, and the ability to work effectively in teams. The standard aims to prepare students for diverse career paths in computing and promote lifelong learning in the rapidly evolving technology landscape [8].

ABET Accreditation Criteria sets the standards for accreditation of programs in applied science, computing, engineering, and technology, focusing on quality, student outcomes and continuous improvement. It was founded in 1932 and has since become a globally recognized accrediting agency for STEM programs. ABET regularly updates its criteria to align with industry needs and advancements in technology. Compliance and maintenance involve self-assessment, peer review, and ongoing improvement efforts by institutions seeking accreditation. The ABET Accreditation Criteria for computing programs emphasize core competencies, such as technical knowledge, problem-solving abilities, teamwork, communication skills, and ethical conduct. Programs must demonstrate continuous improvement and alignment with industry standards through curriculum design, assessment methods, faculty qualifications, and resources. Accreditation signifies that a program meets recognized quality benchmarks and prepares graduates for successful careers in computing-related fields [9].

ISTE Standards for Educators provide guidelines for educators in integrating technology effectively into teaching and learning, fostering digital citizenship, and promoting innovation in educational practices.The standards were first introduced in 1998 and have since evolved to reflect advancements in educational technology and pedagogy.The ISTE Standards for Educators focus on areas, such as facilitating learning with technology, designing digital learning experiences, promoting digital citizenship, and engaging in professional growth and leadership. Educators are encouraged to integrate technology tools, resources, and strategies effectively to meet diverse student needs, enhance collaboration, and foster critical thinking and creativity in the digital age [10].

CSTA K-12 Computer Science Standards define learning objectives and outcomes for K-12 computer science education, covering foundational concepts, computational thinking, programming skills, and ethical considerations. The standards were first introduced in 2011 and have been updated periodically to reflect advancements in computer science education and emerging technologies. Compliance and maintenance involve curriculum development, teacher training, and assessment practices, aligned with the standards. The CSTA K-12 Computer Science Standards emphasize fundamental concepts, such as algorithms, data representation, programming logic, and computational problem-solving. They also address broader skills, such as collaboration, communication, and ethical decision-making, in the context of technology use [11].

Despite the fact that the area of software testing is considered a massive theoretical and technical branch of the industry, containing a unique set of practices, tools and approaches, its perception is still not fully defined and fluctuates between the "one of competences of a software engineer" and "a completely independent specialization within the industry". A similar trend is typical of the above-mentioned educational standards, which look at software testing from different angles. Some of them may not have specific sections, dedicated solely to software testing and test automation, while they still recognize the importance of these concepts within the broader context of computer science education, software development practices, and technology integration in education:

– while the ISTE Standards for Educators primarily focus on technology integration in teaching and learning, they indirectly recognize the importance of software testing and test automation in educational contexts. Educators are encouraged to promote critical thinking, problem-solving, and digital citizenship, which includes also discussions on software quality, testing strategies, and the role of automation in ensuring reliable technology use.

– the CSTA K-12 Computer Science Standards may include concepts related to software testing and test automation within the broader context of computational thinking and programming skills. Students may learn about the importance of testing code for correctness and reliability, understand basic testing techniques, and explore the potential of automation in streamlining testing processes.

Among the standards mentioned, the ACM/IEEE Computer Science Curricula and ABET Accreditation Criteria specifically highlight quality assurance and software testing as separate areas or skill sets within the broader context of software engineering and provide the recommendations for teaching them in higher education institutions:

– ACM/IEEE Computer Science Curricula acknowledges quality assurance and software testing as distinct knowledge areas within software engineering. It recognizes the importance of QA processes, testing methodologies, and test automation techniques in ensuring software reliability, functionality, and performance. Students are expected to develop skills in designing test cases, conducting testing activities, and utilizing testing tools. The curriculum include topics, such as test planning, test execution, defect tracking, and test reporting, emphasizing the significance of QA and testing in the software development lifecycle.

– ABET's accreditation criteria for computing programs include specific expectations related to quality assurance and software testing. It considers testing as a critical skill set for software engineers, highlighting the ability to design, implement, and execute test cases, as well as analyze test results and improve software quality. The criteria also address topics, such as test automation, regression testing, integration testing, and verification/validation techniques. ABET emphasizes the integration of QA and testing principles throughout the curriculum to prepare graduates for roles that require expertise in ensuring software quality and reliability.

In the USA, universities offering IT specialties are not required to conform to specific standards, such as the ACM/IEEE Computer Science Curricula, ABET Accreditation Criteria, ISTE Standards for Educators, or CSTA K-12 Computer Science Standards. However, the majority of them choose to align their IT programs with one or a set of these standards for several reasons:

1. Accreditation: ABET accreditation is highly respected in the field of engineering and technology. Many universities seek ABET accreditation for their IT programs as it demonstrates that the program meets recognized quality standards.

2. Industry Relevance: aligning with standards, such as the ACM/IEEE Computer Science Curricula, ensures that the curriculum is relevant to industry needs and prepares students with the necessary skills and knowledge for their careers, which expands the opportunities for cooperation between universities and the information technology industry.

3. Pedagogical Best Practices: standards like the ISTE Standards for Educators and CSTA K-12 Computer Science Standards provide practical guidelines for effective teaching and learning practices in technology-related fields.

4. Competitive Advantage: Universities that adhere to recognized standards may have a competitive advantage in attracting students and faculty, contributing to its development and expansion.

While conformity to these standards is not mandatory, they serve as valuable benchmarks and guidelines for universities to enhance the quality and relevance of their IT programs. It worth mentioning that universities may also have their own internal standards and guidelines that align with or supplement these broader standards.

The different approaches of educational standards to the perception of testing are also reflected in the differences in the training programs themselves at universities in the United States of America. One of the most prestigious universities in the country is the Massachusetts Technical University, which, of course, offers a range of both undergraduate and master's degrees in software development [12]. The Bachelor in Computer Science and Engineering at MIT is typically a four-year full-time program. Admission to this program is highly competitive, requiring a high school diploma or equivalent, with a strong background in mathematics and science. Beyond formal coursework, students have access to a wealth of informal educational opportunities, including participation in research projects, internships, hackathons, and industry collaborations. To graduate, students must complete core computer science courses, a selection of elective courses, and a capstone project, ensuring they have both the theoretical knowledge and practical experience necessary for success in the field of software development. In general, the curriculum covers aspects of software engineering, quality assurance, and testing methodologies, providing students with a comprehensive understanding of these critical areas [13]. However, testing is not put into a separate discipline or training program but considered as only one of the topics in the context of the broader disciplines. For example, the course “6.100 – Introduction to Computer Science and Programming” covers a wide range of information technology fields, such as computation, Python language scripting, simple algorithms and data structures, debugging, and algorithmic complexity, while also including modules of quality assurance and testing principles [14]. Similarly, technology-specific discipline programs, such as “6.1820 – Mobile and Sensor Computing” and “6.1060 – Software Performance Engineering”, cover testing processes in the areas of mobile development and performance engineering respectively, and offer specific tools and practices that are more relevant to these areas [15, 16].

A similar situation can be observed for another popular and prestigious educational institution of United States: Stanford University, which also offer the Bachelor in Computer Science training program [17]. There are also no separate courses for software testing in the subjects studied. However, problematics of quality control, methodologies and testing tools are exposed both in a wide-range disciplines, such as “CS 107 – Computer Organization and Systems”, and in specific technical or organizational areas, for example, “CS 194 – Software Project Management” [18, 19].

One example of a university that has a separate discipline focusing on software testing and quality assurance in its educational program is Carnegie Mellon University [20]. In the course of the Bachelor of Computer Science programs, this university is not fundamentally different from those mentioned above (Stanford and MIT), offering only disciplines that consider testing in the context of more global areas of knowledge, e.g., “15-413 – Software Engineering” and “15-410 – Software Architecture and Design”. However, there are significant differences in the Master of Computer Science programs, in which students can choose from a wide range of disciplines directly related to both the general problem of software testing and its individual technical or organizational aspects. “17-323 – Quality Assurance” is one of the fundamental disciplines in a context of software quality assurance and control. This course introduces various testing tools and techniques to software engineering students, developing the knowledge of when those tools should be used, how to evaluate their results, and what assurances they can provide. The key learning objectives of the course include:

1. Understanding of software quality: how to define it, analyze it, and measure it.
2. Selecting the proper analytical tool/technique for a given situation.
3. Understanding of the strengths and weaknesses of different quality assurance techniques, such as software testing, static analysis, code review.
4. Learning of how to collect, manage, and evaluate quality metrics.
5. Analysis and verification of a variety of software properties including security, reliability, and performance.

“17-443 – Quality Management” is another testing-related discipline, introducing students to the managerial challenges of developing high quality software systems. The key learning objectives of this course include:

1. Definition of a quality management process in the context of a software project.
2. Understanding the costs, associated with achieving quality goals and not achieving them.
3. Understanding the tradeoffs, required to implement quality assurance techniques.
4. Gaining an experience of using the quality metrics to inform project-level decisions.

"15-414 – Bug Catching: Automated Program Verification" deserves a special mention, as it is dedicated to the problematics of automated testing. The course teaches students how to write bug-free code through the process of software verification. Students learn the principles and algorithms behind automated verification tools and understand their practical limitations while gaining experience writing verified, machine-checked code [21, 22].

Speaking of the limitations of this study, it is worth noting the limited access to materials of some educational programs, as well as the advisory nature of the approaches and methods, described in them: not all educational materials, the content of lectures, workshops, and extra-auditorium work, are available in the public space and can be analyzed without the participation of representatives of the appropriate educational institutions directly. It is also worth noting that the analysis examined only a few of the most representative institutions in the United States of America, which does not give a full assessment of the overall approach to the training of testers across the whole country.

As has already been mentioned, although the USA plays a key role in the development of the international IT sphere, it is not the only technology ambassador. Key universities in Europe and Asia are also able to offer their unique expertise in training software testing engineers, which opens prospects for further research.

6. Conclusions

An analysis of the experience of training of software engineers in foreign institutions of higher education, discussed above, allows to highlight the following trends in curriculum formulation and the organization of the educational process in the leading technical universities of the United States of America according to the defined objectives of the study.

1. The existence and independent development and implementation of several nationally accepted standards of training in the field of information technology in higher education institutions. On the one hand, it gives the universities the space to adapt the educational material to the particular educational process of a particular institution and to implement their own vision of the training of future professionals. On the other hand, it is worth noting that despite differences in interpretation and emphasis on different theoretical and practical specifics of some disciplines, the content of the educational standards, considered in the article, is mostly comparable and repeating each other, which implies stability in the set of theoretic knowledge and the quality of training of specialists regardless of the orientation of the university to a certain standard.

2. Despite some differences in the content of the training programs, all without exception make the initial emphasis on the basic areas of knowledge, such as algorithms, programming languages, software development, databases and computer architecture, as well as fundamental disciplines, such as physics, mathematics, the foundations of algorithms, which allows to form a theoretical basis for further study of specialized disciplines. It should be noted that the place and status of the software testing in the general IT industry is not fully defined either in educational standards or in training programs: quality control and testing area in general can be either distinguished as separate disciplines and specialties, or considered as an appendix to the material of broader technical areas, such as software development. However, it is worth noting that the problematics of software testing in any case stands out as an integral part of the competence of future professionals in the field of information technology, and regardless of the availability or absence of specialized disciplines, students of the specialty "Computer Science" of all the universities acquire such skills as development of test scenarios, conducting testing activities, use of testing tools, tracking defects and others.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its resultsб presented in this article.

Funding

The study was performed without financial support.

Data availability

Manuscript has no associated data.

Use of artificial intelligence

The author has used artificial intelligence technologieswithin acceptable limits to provide his own verified data, which is described in the research methodology section.

Acknowledgments

The author would like to express his sincere gratitude and deep respect to the editorial staff of this journal, who is working in one of the most dangerous and war-affected regions of Ukraine, as well as to the defenders of Ukraine who are fighting for the freedom and independence of our country. Thanks to your efforts, the Ukrainian scientific community continues its work and development.

References

1. Avsheniuk, N., Diachenko, L., Kotun, K., Marusynets, M., Ohiienko, O., Sulyma, O., Postryhach, N. (2017). Zarubizhnyi dosvid profesiinoi pidhotovky vchyteliv. Kyiv: DKS "Tsentr", 83.

2. Vorobiova, O., Debych, M., Luhovyi, V., Orzhel, O., Sliusarenko, O., Talanov, Zh., Tryma, K. (2019). Analiz providnoho vitchyznianoho ta zarubizhnoho dosvidu otsiniuvannia yakosti vyshchoi osvity v konteksti yevropeiskoi intehratsii: Analitychni materialy (Chastyna II). Kyiv: Instytut vyshchoi osvity Natsionalnoi akademii pedahohichnykh nauk Ukrainy, 150.

3. Ctriuk, A. M. (2018). Software engineering: first 50 years of formation and development. CEUR Workshop Proceedings, 2292, 11–36.

4. Schachterle, L., Demetry, C., Orr, J. A.; Patil, A., Gray, P. (Eds.) (2009). Quality Assurance in Engineering Education in the United States. Engineering Education Quality Assurance. Boston: Springer. https://doi.org/10.1007/978-1-4419-0555-0_13

5. Garousi, V., Mathur, A. (2010). Current State of the Software Testing Education in North American Academia and Some Recommendations for the New Educators. 2010 23rd IEEE Conference on Software Engineering Education and Training. https://doi.org/10.1109/cseet.2010.29

6. Astigarraga, T., Dow, E. M., Lara, C., Prewitt, R., Ward, M. R. (2010). The Emerging Role of Software Testing in Curricula. 2010 IEEE Transforming Engineering Education: Creating Interdisciplinary Skills for Complex Global Environments. https://doi.org/10.1109/tee.2010.5508833

7. Melo, S. M., Moreira, V. X. S., Paschoal, L. N., Souza, S. R. S. (2020). Testing Education. Proceedings of the XXXIV Brazilian Symposium on Software Engineering. New York, 554–563. https://doi.org/10.1145/3422392.3422483

8. Curricula Recommendations. ACM/IEEE Computer Science Curricula. ACM and IEEE. Available at: https://www.acm.org/education/curricula-recommendations

9. Accreditation Criteria & Supporting Documents. ABET. Available at: https://www.abet.org/accreditation/accreditation-criteria/

10. ISTE Standards for Educators. International Society for Technology in Education. Available at: https://www.iste.org/standards/ iste-standards-for-teachers

11. CSTA K-12 Computer Science Standards. Computer Science Teachers Association. Available at: https://www.csteachers.org/ page/standards

12. Massachusetts Institute of Technology. Available at: https://web.mit.edu

13. Computer Science Computer Engineering Track 2023-2024. Stanford University. Available at: https://drive.google.com/file/d/1BTFWYpVkGlaBF3vigGXVifxwY7qLoPwh/view

14. 6.1020: Software Construction. Massachusetts Institute of Technology. Available at: https://catalog.mit.edu/search/?P=6.1020

15. 6.1820: Mobile and Sensor Computing. Massachusetts Institute of Technology. Available at: https://catalog.mit.edu/search/?P=6.1820

16. 6.1060: Software Performance Engineering. Massachusetts Institute of Technology. Available at: https://catalog.mit.edu/search/?P=6.1060

17. Academics: Bachelor's Program. Stanford University. Available at: https://www.cs.stanford.edu/academics/academics-bachelors-program

18. CS 107: Computer Organization and Systems. Stanford University. Available at: https://web.stanford.edu/class/cs107/

19. CS 194: Software Project. Stanford University. Available at: https://web.stanford.edu/class/cs194/

20. Bachelor of Science in Computer Science. Carnegie Mellon University. Available at: https://csd.cmu.edu/academics/bachelors/overview

21. CS Sample Curriculum Schedule. Carnegie Mellon University. Available at: https://csd.cmu.edu/sites/default/files/2024-04/CS_Sample_Curriculum_Schedule.pdf

22. Computer Science Program. Carnegie Mellon University. Available at: http://coursecatalog.web.cmu.edu/schools-colleges/schoolofcomputerscience/undergraduatecomputerscience/#bscurriculumtextcontainer

Received date 09.04.2024
Accepted date 21.05.2024
Published date 31.05.2024

Oleksandr Gura, PhD, Department of General and Applied Physics, Zaporizhzhia National University, Zhukovskoho str., 66, Zaporizhzhia, Ukraine, 69600
E-mail: ol.ol.gura@gmail.com