YEARBOOK OF UNWE (2024), ACADEMIC PUBLISHING HOUSE – UNWE
ISSN (print): 1312-5486; ISSN (online): 2534-8949; http://unwe-yearbook.org/en/
DOI: https://doi.org/10.37075/YB.2024.2.06

QUALITY ASSURANCE METHODS OF SOFTWARE APPLICATIONS AND ONLINE SERVICES IN INSTITUTIONS OF HIGHER EDUCATION

Elitsa Pavlova[1]
e-mail: epavlova@unwe.bg

Abstract

The report examines methods for ensuring the quality of software applications and online services in higher education institutions (HEIs). It highlights key challenges related to security, system complexity, and user diversity. Testing approaches such as performance, compatibility, security, and usability testing are analysed. It emphasizes automation, integration with DevOps processes, and the use of standardized tools. The report offers solutions for reducing vulnerabilities and ensuring software reliability and functionality. The main conclusion is that continuous testing and integrated security are critical for data protection and the smooth operation of university systems.

Keywords: quality assurance, software applications, online services, security, testing methods, institutions of higher education, cyber threats, integration with DevOps

JEL: O10, O14

Introduction

Institutions of Higher Education (IHE) build, maintain and develop a large information infrastructure to create, store, use and distribute educational resources. Security is critical to quality assurance and is key to the smooth operation of all software applications and online services, including web platforms, sites and APIs. Good practices emphasize that security should be included throughout the software development life cycle. Effective quality assurance testing is all about finding missed bugs and threats, using standardized testing approaches. Regular testing and updating of software and prioritization of risks based on their impact are necessary to ensure that University information systems and assets are protected against the latest threats. Higher education policies do not directly affect innovation, but they do have a link with the drivers of innovation, according to a study of EU Universities (Baldwin, 2021).

Cyber Threats to Education Report Documents High Level of Vulnerability to Cyber Attacks in Academic Environments (Fokker, 2021). IHEs are vulner-

[1] Eng., PhD, Department of National and Regional Security, Faculty of Economics of Infrastructure, University of National and World Economy, Bulgaria

able to cyber-attacks due to their decentralized structure, diverse group of users, operation of outdated information systems, multiple communication devices and sensitive data. A look at the World Higher Education Database shows that 63% of universities manage the enrolment and storage of student data entirely online (WHED, 2020).

Software applications and online services

Universities provide numerous administrative services electronically for students, PhD students and teachers. Visibility of these services and applications, traffic to them, and endpoint connectivity are critical. University sites have specific functionalities in the areas of user management, service catalogue, content creation, data filtering, source code modification capabilities, etc. Access requires identification with a username and password, which are often used to access other university information systems.

The dynamic threat landscape and the large number of applications and web services in the IHEs make manually updating, testing and deploying security policy sets and controls an impossible task. Cyber Threats to Education Report Documents High Level of Vulnerability to Cyber Attacks in Academic Environments (Fokker, 2021). IHEs are vulnerable to cyber-attacks due to their decentralized structure, diverse group of users, operation of outdated information systems, multiple communication devices and sensitive data. A look at the World Higher Education Database shows that 63% of universities manage the enrolment and storage of student data entirely online (Benavides et al., 2020). This is precisely why tools and models are needed that can provide automation and enable orchestration across the entire application infrastructure. It is important that the web application security tools and controls fit their processes and integrate with the tools that the department's quality assurance teams use. University sites are built as portals and lead to multiple sub-sites (sub-domains) on the Internet or Intranet. Subdomains provide information about basic organizational units and are used to share resources between different departments or directorates of the university. Software products should have a wide range of features and capabilities, security by design, embedding testing into all processes, and support for new technologies (Toptal Insights, 2022).

Priority areas for each university are administration, candidate-student campaign, e-learning, web student, information security, archiving and storage, creation of websites and mobile applications, library system and catalogue and communication and cooperation. For the activities, wherever possible, ready-made popular software developments are used. Examples of such are:

- University mail. Popular platforms: Microsoft Office 365 Exchange, Web mail, Google mail.
- E-learning – software products and systems providing opportunities for synchronous and asynchronous learning, as well as remote authorized access of students, teachers and administrators to activities, resources and systems. Popular platforms: M-Learning, E-Learn, Moodle, Teams.
- Communication and cooperation - services related to communication and cooperation between members of the university community. Popular platforms: Confluence, Microsoft Teams, Zoom, Drupal, Instant Messaging, ServiceNow, and Google Forms.
- Information security – software products related to the security of information assets, network connectivity, disk encryption, phishing reporting, SSL certificates, Wi-Fi wireless network, remote access, etc. Popular Platforms: Network Security Monitoring, Encryption, Web Vulnerability Scanning, Network Defence Wireless, Packet Sniffers, Antivirus Software, Firewall, PKI Services, Managed Detection Services, Penetration Testing
- Backup and storage – software products and systems related to file and data backup, file and data recovery, automated access solutions, etc. Popular platforms: OneDrive Cloud Storage, Qualtrics, Site-Licenses, Google Workspace, SSC Guide.

Website creation tools – open-source software products and content management systems. Popular platforms: WordPress, TYPO3, Joomla, Drupal, Contao, Neos, WooCommerce, OpenCart, AbanteCart, PrestaShop (Opensource.com, 2023).

IHEs have many specific activities for which it is necessary to develop their own software or information systems. The methods outlined in the report are highly applicable across a variety of university sectors. Specific examples of their applicability include:

Administration – software products and systems for supporting the life cycle of the educational process, document issuance, settlement of student status, fulfilment of student obligations, record-keeping and accounting system, system for electronic tracking of document circulation, study plans, research administrative systems, storage of scientific data and publications, technical support, etc.

Candidate-student campaign – software products and systems related to the application and admission of students to the Professional Bachelor's College, the Bachelor's College, the Master's College and doctoral students. Student identification system for the electronic conduct of examinations and evaluations held in specialized test centres.

Web student – a university-specific information system for the administration of the educational process and connection with web platforms for dormitories and scholarships. Through them, electronic application and ranking for scholarships and dormitories is carried out, using the web student database.

Library system – online catalogue of the university library with textbooks, books. Usually, the system has its own developed software and is linked to world-renowned full-text reference databases to which the library subscribes.

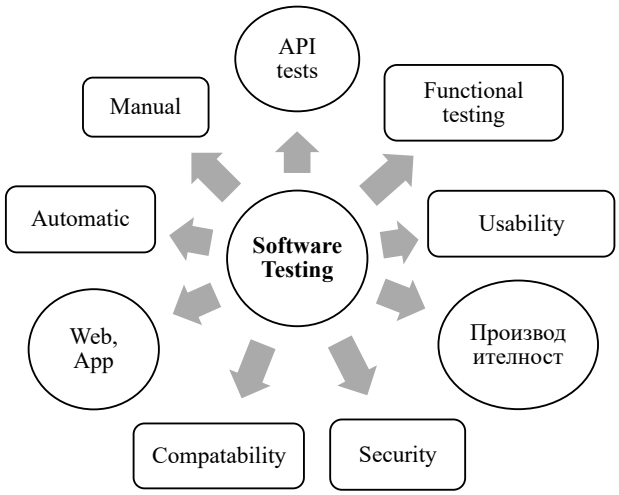
E-learning platforms: Performance and security testing of platforms such as Moodle or Teams is essential to ensure that these services can handle large numbers of students, especially during exam periods. Load testing ensures that these platforms can handle concurrent logins and data transfers without crashes.

Research Data Management: In research-oriented institutions, data security and archiving systems must undergo rigorous quality assurance to ensure compliance with data retention policies and prevent data breaches. Automated security scans can be integrated with cloud storage solutions (e.g., OneDrive or Google Workspace) to provide real-time security monitoring.

The report „Software Testing Tools and Techniques for Web Applications" points out that software products and web services must pass a series of tests to ensure that users and processes will only access information or resources to which they are entitled (Shikha, Bahl, 2023). Quality assurance testing includes all functions designed to control the flow of information and the use of resources by users, processes, and objects. Software products associated with administrative systems must have requirements to provide specific functions designed to ensure that data will not be modified in an unauthorized manner. Fault detection and recovery tests minimize interruptions or loss of service.

Quality assurance methods

The most important features of modern online service security tools include visibility and protection, coverage of different architectures, integration with DevOps tools, automation and continuous updates. The purpose of testing is to check whether the specification, program code and documentation created during the development process conform to the rules and standards of the university. By means of the dynamic method, the properties of the web platform were analysed and evaluated according to the results of the real work. The testing approaches are the same for websites and web services as well as for APIs. The difference is that web services are available on servers, but for API servers, browsers and URLs are optional because APIs can be accessed locally (Bhambhu and Srivastava, 2009). The most used tests of software applications and online services are shown in Figure 1.



Source: Bhambhu and Srivastava (2009).

Figure 1: Types of testing of software applications and online services

Support testing. Testing is performed on software that has already been implemented during an upgrade, change, or migration cycle to other hardware. Testing can be divided into two types: confirmation testing to ensure that bugs and errors previously discovered have been fixed and regression testing to verify that previously developed and tested software still works without errors after it has been changed.

Performance testing. This kind of testing is very necessary to verify that service level agreements are met for APIs. It gives an idea of how the website will react to a load with a certain number of visitors. The use of the test bed excludes the possibility of overloading and ensures continuous operation during peak loads. Testing can be done by loading scripted tests to any API performance tool and running simultaneously with a larger number of users.

Usability testing. It is performed to determine whether the created application is easy to use. It includes user interface testing to check the efficiency of navigation and the degree of user perception of information and cross-platform testing to check its suitability to work on multiple platforms and different operating systems and devices.

Code review and quality analysis. Systematic review of the software source code to detect and correct errors that went unnoticed in the initial development phase to improve the quality of the software.

Compatibility testing. A type of software testing used to ensure product compatibility with various other entities such as web browsers, platforms, users (in cases of very specific types of requirements), etc. This type of testing helps determine how successful the system's performance is in a specific environment that includes hardware, network, operating system and other software, etc.

Security testing. It is done to ensure that APIs are built in a secure manner and are protected from any vulnerabilities and malicious attacks. Security testing includes user authentication and authorization, data leakage, injection of vulnerabilities such as XSS, SQL injection. Tools like Postman, Karate, fiddler, JMeter are used to test API security (Ehsan et al., 2022).

Quality assurance software tools

Functionalities and standards that web service security analysis software products typically check include conformance validation, integrity checks, XML schema validation, XML encryption, XML signature, WSSecurity, user authentication, auditing, alerting, access control web services and content verification (PractiTest, 2023). The most used tools for analysing source code and identifying security weaknesses divided by category are:

- Test management tools, which are used to store information about how testing should be performed, plan test activities, and report the status of quality assurance activities. Examples of such tools are Microsoft Test Manager, Jira, Zephyr, etc.
- Software review tools that are used to improve code quality. Examples of such tools are Gerrit, Crucible, Phabricator, and Bitbucket.
- Bug tracking tools used to manage and track software bugs and issues during development. One of the most popular tools is Github Issues, through which developers can assign tasks, categorize bugs by type, add tags, and set milestones to track progress.
- Continuous integration tools that automate the process of building, testing and deploying software applications. One of the most popular tools in this category is GitHub Actions, which allows developers to automate software workflows, including continuous integration and continuous deployment processes.

Initial security assessments should be performed as early as possible in service design to ensure that security controls and corrective actions are performed in the most efficient and cost-effective manner. A specific type of testing is conducted after functional testing is completed to ensure that user requests do not change the information system and service in a way that affects the security posture or the application of required security controls.

A bug report is a written document describing a particular bug found during a specific phase of the testing process (Vijay, 2023). Provides detailed problem information, helps maintain an archive for future reference, categorizes bugs to aid in root cause analysis, and prevents duplicate problem reporting. The report should contain a brief statement of the problem, a detailed description of the problem, input data, expected and actual results, test environment, URLs and screenshots. Each reported bug is assigned a severity and priority. The risks, costs, opportunities, and benefits of fixing or not fixing the error are also determined. A 2021 software security report notes that more than 85% of web applications have security vulnerabilities. Understanding these vulnerabilities is key to detecting them and protecting data. Analysis of the report shows that the probability of detecting a critical error in a dynamic web application is about 35% through automatic scanning and 96% through comprehensive expert analysis. Administrative problems are a 20% more common cause of vulnerability than system development errors. Security assessment and management processes ensure that all critical assets are protected, and risks are adequately mitigated.

The report provides a comprehensive overview of quality assurance (QA) methods used for software applications and online services in higher education institutions. The main objective is to examine how universities, with their diverse systems and services, can ensure the reliability, security and functionality of the software systems that support both academic and administrative activities. The findings highlight several critical aspects:

- Variety of software and service types: Universities rely on a wide range of software applications to support administrative services, e-learning, information security and communication. These applications are crucial for managing student data, e-learning resources, communication tools and library systems.

Security vulnerabilities: Universities are vulnerable to a range of cyber threats due to their decentralized structure and the sensitive nature of the data they manage. The report highlights the importance of integrating security into every stage of the software development lifecycle (SDLC) to mitigate risks (Alothman et al., 2022).

Comprehensive testing and continuous monitoring: The dynamic and rapidly evolving nature of the threat landscape requires continuous testing and updates. This includes using standardized testing approaches such as performance, security, and usability testing to ensure software quality and security.

- Automation and integration: The need for automated security and testing tools is highlighted, especially given the large volume of software and services used in IHE. Integrating these tools into DevOps workflows is critical to ensure continuous quality assurance. As more universities move to online

enrolment and storage of student data, compliance with data privacy regulations such as GDPR becomes critical. Institutions must ensure that their QA methodologies not only address functional quality, but also maintain the integrity and confidentiality of sensitive data.

The report highlights the importance of DevSecOps practices, where security is integrated into the DevOps pipeline from the start (Alothman et al., 2022). This shift to a security-first approach is essential to protect both academic and administrative systems against evolving cyber threats. A recurring problem in universities is decentralized software development. Often, academic units or faculty members create software without consulting the central IT department, leading to security vulnerabilities. This lack of coordination exacerbates the risk of breaches, especially when the software does not have standardized security features. IHEs have complex roles and access controls due to the diversity of users (students, faculty, administrative staff). This complexity requires a more granular approach to user authentication, authorization, and access control, which can be addressed through robust quality assurance testing techniques such as security testing and user behaviour analysis.

Conclusion

Major problems in ensuring the quality of software products, web services and APIs are determined not only by the functions of individual information systems, but also by many other factors. In a university there are different groups of users and complex dependencies between them. Changing the roles of learners in the system requires careful refinement of the individual subsystems. Individual information systems and subsystems have a different approach to information security, some are closed systems of the type of financial and accounting activity, while others are completely open and intended for research activity. Roles in IT departments are very diverse, and one job often combines many roles. Sometimes individual academic units and teachers create software products, web services without consulting the IT department, and this often leads to a breach of information security. As a result, control over the processes and the construction of the software products is associated with many competing requirements.

To further enhance the originality and applicability of the study, the following innovations can be explored: AI-powered Security Testing, blockchain for Academic Integrity, student-led quality assurance initiatives, inter-institutional Quality Assurance Framework. Incorporating AI tools into vulnerability detection can improve the effectiveness of security testing. AI can automatically detect anomalies in real-time traffic on university platforms, providing proactive alerts of potential breaches or weak spots. Blockchain technology can be deployed to create

secure, transparent records of academic achievements and sensitive transactions such as enrolment, graduation, and scholarship disbursement. Integrating this into the quality assurance framework can improve data integrity and security in IHE. Encouraging students to participate in the quality assurance process through internships or collaborative projects can not only enhance hands-on learning, but also enhance the capacity of IHEs to manage their growing software portfolio. This can be achieved by using gamified security training programs, where that students learn through simulated real-world attacks. Developing a model for inter-institutional collaboration to standardize quality assurance processes can help IHEs share best practices, tools, and methodologies. By applying these methods to various university operations, institutions can better manage their software environments, reduce risks, and improve the overall quality of services provided to students and faculty.

References

Opensource.com. (2021). 3 open source content management systems compared, available at: https://opensource.com/business/14/6/open-source-cms-joomla-wordpress-drupal.

Alothman, B., et al. (2022). Developing a Cyber Incident Exercises Model to Educate Security Teams, Electronics, 11(10), p. 1575, https://doi.org/10.3390/electronics11101575

Benavides, L.M.C., et al. (2020). Digital Transformation in Higher Education Institutions: A Systematic Literature Review, Sensors, 20(11), p. 3291, https://doi.org/10.3390/s20113291

Bhambhu, L. and Srivastava, D. (2009). Testing technology, International Journal of Information Technology and Knowledge Management, Volume 2, No. 1, pp. 145-148.

Baldwin, R.G. (2021). Technology in Higher Education, available at: https://education.stateuniversity.com/pages/2496/Technology-in-Education-HIGHER-EDUCATION.html

Ehsan, A., et al. (2022). RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions, Applied Sciences, 12(9), p. 4369, https://doi.org/10.3390/app12094369

Fokker, J. (2021). Cyber Threat Intelligence Reports, Trellix, available at: https://www.trellix.com/en-us/advanced-research-center/threat-reports/jul-2022.html

PractiTest. (2023) Best QA Testing Tools, available at: https://www.practitest.com/best-testing-tools/

Shikha, M., Bahl, K. (2023). Software Testing Tools & Techniques for Web Applications, International Journal of Engineering and Technical Research

(IJETR), Volume 3, Issue 5, available at: https://www.academia.edu/25840184/Software_Testing_Tools_and_Techniques_for_Web_Applications

Toptal Insights. (2022). Cybersecurity in Higher Education: Problems and Solutions, Toptal Insights Blog, available at: https://www.toptal.com/insights/innovation/cybersecurity-in-higher-education

Vijay. (2023). How to Write a Good Bug Report?, Software Testing Help, available at: https://www.softwaretestinghelp.com/how-to-write-good-bug-report/

WHED. (2020). World Higher Education Database (WHED), available at: https://www.whed.net/home.php