SOFTWARE QUALITY IN UNIVERSITY TEACHING

Ljubica Kazi

Abstract. *Aim of this paper is to present the status of university teaching in the area of software quality. The concept of software quality is derived from appropriate ISO standards, but it is also related to other characteristics. University teaching development is encouraged by both research results and industry demands. This paper presents literature review of scientific work related to university teaching of software quality. Empirical study has been conducted with analysis of job advertisements, related to human resources demands in software industry, with particular emphasis on software quality knowledge and skills requirements. Second empirical study is related to software quality content included in university teaching – supported by particular study programmes, curriculum of study programmes with teaching subjects and topics within syllabus of particular courses. This paper also contributes with educational framework that integrates different aspects of software quality, with particular emphasis on quality of software as a product and code quality.*

Key words: software quality characteristics, ISO standards, software industry human resources demands, university teaching content, educational framework.

Introduction

Higher education, as being in synergy with research and industry, continually improve teaching content, methods and resources in aim to support the needs for appropriate knowledge and skills of future employees. In aim to improve higher education and align it with the needs of industry and research results, the concept of quality and innovation with constant collaboration with companies has been proposed to be included in educational process [1]. Constant transformation of educational process is inevitable and in that evolution, particularly important was teaching shift from content and task-oriented approach towards students' skills-centered approach, i.e. gaining appropriate teaching and learning outcome [2]. In

[3], teaching quality in higher education has been in focus, with introduction of process-oriented T-CMM (teaching capability maturity model). The importance of improving educational content in study programmes, i.e. curriculum, particularly for software engineering has been addressed in [4].

Software quality as a concept has been addressed as an important part of software development projects, with special emphasis on process and roles that influence the product. Particular aspects include standards and quality assurance methods and tools application. This paper presents results in three aspects of software quality in higher education context – literature review of research results, industrial needs for particular knowledge and skills presented in job advertisements and educational support in study programmes curriculum and subjects syllabus.

The content of this paper is organized as follows. Section two presents theoretical background explaining basic terms related to software quality. Section three presents research methodology, with particular focus on methods and sample for literature review and empirical research. Section four presents results in literature review with two subsections – statistical analysis of published papers and descriptive literature review. Section five provides results in empirical research – regarding study of software industry needs, as well as analysis of software quality educational contents at universities. Section six provides the proposed educational framework to teach students in area of software quality. Having all these results, the appropriate conclusions are drawn in the final section, as well as future research and practice directions.

Theoretical background

Initially, the concept of software quality could be related to process aspects in project management, international standards related to software development artifacts, software as a product and software in use, as well as to the software quality assurance, primarily with software testing in focus.

The process aspect of software quality includes the development process in a broader context of project management, where quality is one of key knowledge areas, parts of project plan and success monitoring. The iron triangle of project success includes scope, time, cost (resources) and quality [5].

During the software development process, many artifacts are created, such as documentation, models and code. Each of them affects the final software product, which influence quality of software in use. All these aspects are addressed in ISO standards, such as ISO 9126 and ISO 25010. ISO/IEC 25010:2011 [6] describes software product quality characteristics as: functional suitability, reliability, performance efficiency, operability, security, compatibility, maintainability, portability. In the same standard, in category of "software quality in use" the emphasize is on effectiveness, efficiency, satisfaction, safety and usability as key software features.

Software development process includes activities in software quality assurance, in all activities [7] and related to every artifact [8]. Software quality assurance includes particular activities in software testing [9], but modern approaches include test-driven development [10] with emphasis on test-based user requirements specification [11, 12].

Research methodology

This paper addresses research results, industry needs, and educational aspects of software quality.

Research results were analyzed from two approaches. First approach is related to statistical presentation of published papers related to the importance of software quality and university teaching of software quality. Sample for this research consist of 350 papers obtained from Google Scholar scientific browser. Second approach is related to descriptive presentation of literature review in context of higher education applications of software quality concept within the teaching contents and methods.

Industrial aspect of software quality has been examined regarding the needs for qualified employees in applying software quality assurance methods and tools. Aim of this empirical research is to determine work positions related to software quality and to extract and categorize knowledge and skills requirements, particularly for QA, project management and software developer positions. Sample for this research consist of job advertisements posted at LinkedIn social network.

Educational aspect of software quality has been examined in this empirical research with extraction and analysis of software quality support in study programmes, courses within study programmes curriculum and topics within the syllabus within particular courses. Sample for this research is formulated by extraction of available data from university schools web portals. First group of university schools are selected from the Shanghai ranking list of universities, while second group of university schools are selected among Serbian state-owned universities.

Literature review

Statistical analysis of published papers

The sample of 350 papers was obtained from Google scholar on 10th November 2022, by using search key words "the importance of software quality" and "university teaching of software quality". The obtained papers could be categorized as having focus on:

- *Key concepts of software quality* – definition of software quality, software quality characteristics, importance of software quality, software quality models and TQM, user impressions on software quality, information system context of software quality, trends in software quality, cost of software quality (value, economics), software quality engineering,
- *Impact of development process to software quality* – DevOps practice, refactoring, branching, code review, code ownership, global distribution, rapid prototyping, software product lines, technical debt, SCRUM, defect tracking, non-technical factors that influence software quality (teamwork, organization, agile methods, developer contribution),
- *Measurements and estimations* – software quality measurements (factor/criteria/metric), estimations and predictions of software defects and quality (using analogia, machine learning, software testing),
- *Application of quality evaluation* – ISO standards application, automated software classification according to software quality level (with artificial intelligence application), quality of artifacts in software development (design models, communication, packages, scenario, components), quality of code and software (clean code, object-oriented design of classes), structural aspect of software quality (design patterns), quality of particular types of software (open source, embedded), automation and tools.
- *Software quality teaching* – education of software engineers (im-

provements of curriculum, mapping of practical needs with software engineering teaching, practical approaches in software engineering teaching, analysis of software quality education and training in academia and industry), teaching software quality assurance in particular types of software (web applications, eLearning, information systems), teaching particular methods in software quality assurance (software requirements inspections, end-user testing, evidence-based software engineering, static testing, teamwork, disciplined software development).

Statistical presentation of number of software quality – related published papers is provided at Figure 1.

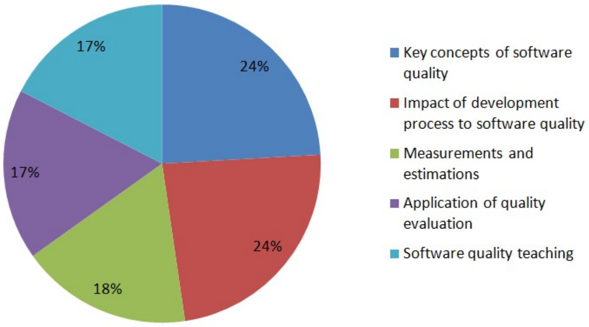


Figure 1. Statistics on published papers on software quality

Descriptive review of literature in educational context of software quality

The need to improve computer science education at universities has been emphasized in [13], particularly in the design and implementation of curricula. This paper addresses software quality as a broad category, that incorporates not only software testing, but other relevant aspects and proposes V- model (including requirements analysis, architectural design, detailed design, coding, unit testing, integration testing, system testing) that could incorporate variety of software quality assurance techniques within a curriculum. Burgess [14] proposed including software quality issues in educational content, starting from the first year of software engineering higher education. Suryn [15] discusses software quality engineering SQE position within the software engineering education (according to Software Engineering Body of Knowledge SWEBOK) and the maturity and stability of quality models, standards, tools that could be used within software quality engineering education. In this paper it has been concluded that the basic methods and tools are immature and the area of software quality engineering should be developed more, to have appropriate material ready for educational process. Suryn emphasizes three core components that should be incorporated in SQE education – 1) quality requirements, 2) quality measurement and evaluation instruments, 3) in-lifecycle quality implementation.

Experiences in teaching software quality assurance as a particular course within software engineering undergraduate curriculum has been presented in [16]. It has been described that topics within this course follow software quality items included in SWEBOK. Special emphasis has been put to tradeoffs in cost-related decision making, while encouraging preventive activities in quality planning. The paper also provided all topics for the theoretical and practical classes in software quality assurance subject (including project plan and requirement specification document, software configuration management, programming, defect management, product quality assessment).

Results of literature review in [17] show that universities have recognized the need for practical experiences to be included in software engineering educational process. This paper recognized several methods for integration of practical demands from industry to regular teaching process: game learning, case studies [18] simulation, inverted classrooms, maintenance projects, service learning etc. Particularly, it is important to prepare students for real working environments, with the use of real-world projects [19, 20] or within the simulation of global software development, which requires specific instructional design and assessment strategies [21]. It is very important to address the challenge of teaching students to practice disciplined software development [22].

Using problem-based learning, this efficient teaching method has been adapted for software engineering students in teaching software quality [23]. Method for teamwork teaching with graduate students has been addressed in [24]. Role playing with development team and software quality assurance team of students (carrying out formal technical reviews of the others' work) has been proposed in [25] as an educational model to simulate real workspace situations, particularly with emphasis on system requirements analysis. In [26], particular specification-based testing has been addressed in teaching environment. Particular focus has been set in [27] for students to perform inspection and evaluation of artifacts (with using check lists), but special focus was put on "specification document".

When software quality is concerned, particular aspects and methods have been addressed in teaching environment, such as teaching software testing concepts using a CODE DEFENDERS game of mutation testing [28]. Paper [29] presents teaching model for end-user testing by using metamorphic method of testing. Static analysis of students' Java programs has been presented in [30]. Evaluation of software architecture quality has been included in the teaching process, by using run-time metrics [31]. Source code quality was in focus in teaching software quality and this analysis was supported by appropriate inspection tool [32]. Particular types of software that are created to support processes in educational institutions have been used for creating a more general quality evaluation model, such as eLearning [33], open-source web applications [34] and Higher Education Institution (HEI) information system [35].

Empirical research results

Empirical study on software industry employees needs

Aim of empirical study on software industry needs was to determine working roles that are explicitly and implicitly related to software quality, as well as to extract the requested and expected knowledge and skills within these roles.

This empirical study has been performed upon available data from job advertisements at LinkedIn social network (data obtained on October 5, 2022). As software quality should be addressed in all software development life cycle activities, extraction of data has been made with appropriate search criteria:

1) Position key words: "Software project management", Location: Serbia;
2) Position key words: "Business analysis", Location: Serbia;
3) Position key words: "Software architecture", Location: Serbia;
4) Position key words: "Software modeling", Location: Serbia;
5) Position key words: "Software design", Location: Serbia;
6) Position key words: "Software development", Location: Serbia;
7) Position key words: "Software quality assurance", Location: Serbia;
8) Position key words: "Software documenting", Location: Serbia.

Results of filtering job advertisements are evaluated and categorized, since the filtering keywords (related to particular software development activities) create output list with job titles that do not always have the working role title precisely related to the keywords-related activity. The initial search results have been presented in Table 1 with the initial number of obtained items, but the more precise working roles obtained from search results have been listed below the Table 1.

Software development activity	**Initial number of job advertisements**
Software project management	144
Business analysis	627
Software architecture	554
Software modeling	21
Software design	698
Software development	1842
Software quality assurance	212
Software documenting	13

Table 1. Results of job advertisements analysis from LinkedIn posts

Having analyzed results of job advertisements search, there could be conclusion that, while having different software development activities in keywords, there are many overlapping, i.e. appearance of the same job advertisements in multiple categories (i.e. search results lists). Therefore, to have further detailed knowledge, experience and skills analysis performed, it is obvious that there should be appropriate distinction, but also merging applied.

Selected job advertisements are those that have appropriate explicit and precise working role title, according to the software development activity. Particular types of work activities with appropriate precise work position titles are listed as:

- Software project or team management – Program manager, Technical product manager, Product Owner, Product Manager, Production support Manager, Lead software engineer, IT project manager, Software delivery project manager, Project Manager, Principal Solutions Consultant, Continuous Improvement Manager, Customer Project Manager, Product Marketing Manager, Agile Team Lead, Principal software engineering manager, Software engineering team lead, Head of group, technical director,
  - Specifics – particular technology-related manager / architect - Manager data analytics, Lead Front-end developer, head of SEO

content, Technical Lead, Mobile lead software architect, Lead Big Data engineer, Embedded software team lead, Data analysis and engineering - data engineer, big data engineer, data scientist, data analyst, data warehousing specialist, internet data evaluator

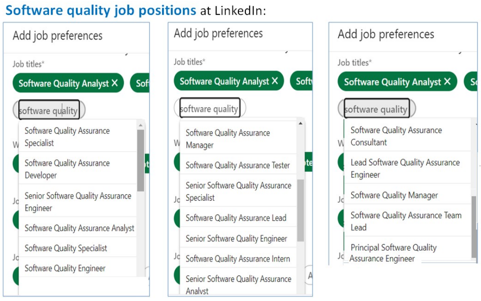
- Business analysis – Analyst, Senior Analyst, System Analyst, Business Operations Analyst, Business Analyst (FinTech/Banking), Financial analyst, Technical requirements engineer and operations, consultant in SAP financial services, digital business analyst/functional consultant
- Software architecture – Web information architect, Senior Solutions Analyst, Custom solution architect, cloud solution architect, cloud and integration engineer, Solution architect, technical consultant, data analytics and automation analyst, process architect, Lead Software Architect, Enterprise Architect, cloud and integration engineer, solution architecture expert, senior integration consultant, chief architect
- Software modeling – financial reporting modeling expert
- Software design – solution designer, digital solutions specialist
- Software development – Senior developer, software engineer, front-end developer, back-end developer, full stack developer, engine developer
  - Specifics – Software developer or consultant for particular technologies and solutions – SAP, Golang, Java, C, C++, back-end engineer, React, unity, java script, php, common backend, python, front-end engineer, Angular, Linux application developer, .NET, Oracle Siebel, Vue JS, cloud data store software development engineer, REST api enginer, Android etc.
- Software quality assurance – QA test automation engineer, senior software test engineer, Quality assurance automation engineer, Software quality assurance tester, quality assurance engineer, QA manual engineer, software quality assurance developer, QA tester, QA engineer, quality engineer, QA analyst, agile quality assurance manager, QA team lead, Lead test automation engineer, engineering manager – quality, test engineer

- Software documenting – Reporting Analyst, financial reporting expert

Other job advertisements, that have also been listed within results in job search with previously mentioned keywords, but their work titles are not precisely related to job categories related to these keywords, could be categorized in areas of:

- Artificial intelligence application – machine learning engineer
- Data analysis – Business intelligence engineer, Data analyst, Datawarehouse engineer, Data Scientist
- Supporting work roles – Software trainer, DevOps Engineer, Software support specialist

Software quality-related working positions (roles) that have been obtained from LinkedIn social network are presented at Figure 2.


Software quality job positions at LinkedIn:

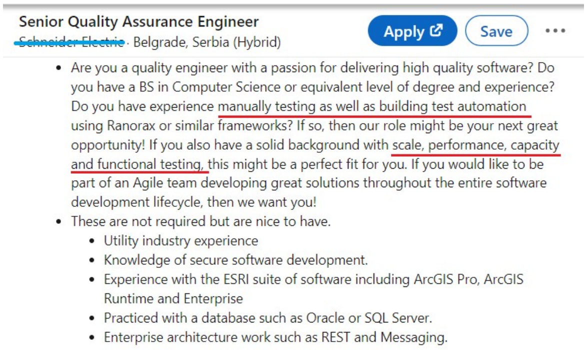
Add job preferences
Job titles*
Software Quality Analyst ✕
software quality
- Software Quality Assurance Specialist
- Software Quality Assurance Developer
- Senior Software Quality Assurance Engineer
- Software Quality Assurance Analyst
- Software Quality Specialist
- Software Quality Engineer

Add job preferences
Job titles*
Software Quality Analyst ✕
software quality
- Software Quality Assurance Manager
- Software Quality Assurance Tester
- Senior Software Quality Assurance Specialist
- Software Quality Assurance Lead
- Senior Software Quality Engineer
- Software Quality Assurance Intern
- Senior Software Quality Assurance Analyst

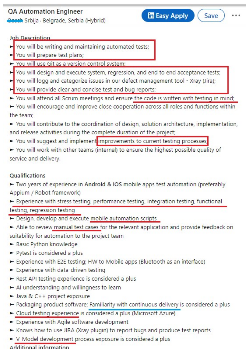
Add job preferences
Job titles*
Software Quality Analyst ✕
software quality
- Software Quality Assurance Consultant
- Lead Software Quality Assurance Engineer
- Software Quality Manager
- Software Quality Assurance Team Lead
- Principal Software Quality Assurance Engineer

Figure 2. The explicit software quality related work positions

Figure 3 presents job advertisements for two software quality-related work positions.


Senior Quality Assurance Engineer
~~Schneider Electric~~ · Belgrade, Serbia (Hybrid)

- Are you a quality engineer with a passion for delivering high quality software? Do you have a BS in Computer Science or equivalent level of degree and experience? Do you have experience manually testing as well as building test automation using Ranorax or similar frameworks? If so, then our role might be your next great opportunity! If you also have a solid background with scale, performance, capacity and functional testing, this might be a perfect fit for you. If you would like to be part of an Agile team developing great solutions throughout the entire software development lifecycle, then we want you!
- These are not required but are nice to have.
  - Utility industry experience
  - Knowledge of secure software development.
  - Experience with the ESRI suite of software including ArcGIS Pro, ArcGIS Runtime and Enterprise
  - Practiced with a database such as Oracle or SQL Server.
  - Enterprise architecture work such as REST and Messaging.


QA Automation Engineer
[redacted] Srbija · Belgrade, Serbia (Hybrid)

Job Description
► You will be writing and maintaining automated tests;
► You will prepare test plans;
► You will use Git as a version control system;
► You will design and execute system, regression, and end to end acceptance tests;
► You will logg and categorize issues in our defect management tool - Xray (Jira);
► You will provide clear and concise test and bug reports;
► You will attend all Scrum meetings and ensure the code is written with testing in mind;
► You will encourage and improve close cooperation across all roles and functions within the team;
► You will contribute to the coordination of design, solution architecture, implementation, and release activities during the complete duration of the project;
► You will suggest and implement improvements to current testing processes;
► You will work with other teams (internal) to ensure the highest possible quality of service and delivery.

Qualifications
► Two years of experience in **Android & iOS** mobile apps test automation (preferably Appium / Robot framework)
► Experience with stress testing, performance testing, integration testing, functional testing, regression testing
► Design, develop and execute mobile automation scripts
► Able to review manual test cases for the relevant application and provide feedback on suitability for automation to the project team
► Basic Python knowledge
► Pytest is considered a plus
► Experience with E2E testing; HW to Mobile apps (Bluetooth as an interface)
► Experience with data-driven testing
► Rest API testing experience is considered a plus
► AI understanding and willingness to learn
► Java & C++ project exposure
► Packaging product software; Familiarity with continuous delivery is considered a plus
► Cloud testing experience is considered a plus (Microsoft Azure)
► Experience with Agile software development
► Knows how to use JIRA (Xray plugin) to report bugs and produce test reports
► V-Model development process exposure is considered a plus
Additional information

Figure 3. Example of two job advertisements in software quality area at LinkedIn

Analysis of job advertisements texts results in extracted knowledge of spoken languages (English, German, Russian) and soft skills that are required: attention to detail, creativity, initiative, ability to work collaboratively in multi-cultural teams, high motivation, fast learning, multi-tasking, high quality standards in own work, strong interpersonal and communication skills, good analytical skills, problem-solving skills, time management and organizational skills, strong working ethic, Strong logical thinking, curious mind, clear communication skills and a pragmatic approach to solving problems, knowledge mining and search skills, positive “Can Do” attitude.

Further analysis of selected job advertisements (those that have work position title align with the software development activity used in search key words) lead to conclusion that here should be merging of results (since activities are closely related and search results lists have similar items). Finally, there are categories for more precise analysis of required software quality-related knowledge, experience and skills. These categories are: management, system/business analysis/documenting, software architecture/design/modeling, software development, software quality assurance. Following are descriptions of core responsibilities and requirements from job advertisements texts (omitting additional knowledge, experience and skills that are mentioned in job advertisements texts).

PROJECT MANAGEMENT POSITIONS

- Responsibilities – project planning, tracking the progress, risk management, preparation of project management documents (project charter, project plan, reports), gather requirements and organize them I backlog with prioritization and schedule, drive product delivery, support users with training and documentation

- Requirements – project management tools, understanding IT architectures, agile methodologies, PMP certificate

QA-RELATED MANAGEMENT POSITIONS

- Responsibilities – cross functional collaboration with technology and innovation departments, develop QA strategy, methodology, performance and framework, help execute vision of the quality, quality and resources planning, lead and grow QA team members, help drive and improve QA in agile environment, focus on continuous QA improvements in usage of appropriate testing tools, test techniques, quality monitoring tools and business cases/logic analysis and recommendations, defining quality metrics, define risks, defining testing processes, leadership, reporting, process improvements, Ccollaborate with the strategic, business and functional stakeholders to ensure the quality metrics are reviewed, closed and agreed upon

- Requirements – knowledge of QA terminology, ITIL framework, ISTQB standards

SOFTWARE QUALITY ASSURANCE POSITIONS

- Responsibilities – Responsibilities related to QA – requirements analysis and correction, execute manual test cases, design and write test cases, development of automated tests, interpretation of testing results, test reports, QA ownership of one or more features, manual and explorative testing, reporting identified defects, tracking and reporting problems and defects using tracking software
- Requirements – Selenium, NUnit, UI testing (Java + Selenide + Allure), API testing

SYSTEM/BUSINESS ANALYSIS/DOCUMENTING POSITIONS

- Responsibilities – business analysis/documenting responsibilities – development and ownership of technical requirements and documentation, acquisition and summarizing customer requirements and business processes analysis, producing technical requirements and user stories, continuous maintenance and improvement of the company requirements knowledge bases, technical writing, system requirements documentation, system functional specifications.
- Requirements – systems analysis, requirements management for user interfaces and backend components, technical writing.

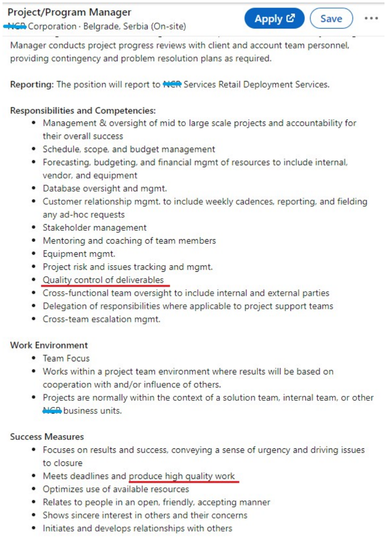
SOFTWARE ARCHITECTURE/DESIGN/MODELING POSITIONS

- Responsibilities – design entire system based on requirements, choose system architecture and components, translate designs and requirements in code, build and organize development workflows to optimize code delivery, write technical specification documents, provide mentorship and guidance to development team, participate in hiring and on boarding of new members
- Requirements – knowledge of web services, software development experience, experience with cloud providers, frameworks, design patterns, data modeling, knowledge of architectural styles and patterns, all aspects of software development life cycle, code versioning systems, ability to analyze and resolve complex technical business issues and technical

problems.

SOFTWARE DEVELOPMENT POSITIONS

- Responsibilities – build and maintain code according to storyboards and use cases
- Requirements – knowledge of modern software development practices and coding standards, agile product development, software development life cycle, databases, programming languages, continuous integration and deployment, automation, architectural and design patterns, asset management, performance management, object-oriented programming, test driven development, regular expressions.



Project/Program Manager

[redacted] Corporation · Belgrade, Serbia (On-site)

Manager conducts project progress reviews with client and account team personnel, providing contingency and problem resolution plans as required.

Reporting: The position will report to [redacted] Services Retail Deployment Services.

Responsibilities and Competencies:

- Management & oversight of mid to large scale projects and accountability for their overall success
- Schedule, scope, and budget management
- Forecasting, budgeting, and financial mgmt of resources to include internal, vendor, and equipment
- Database oversight and mgmt.
- Customer relationship mgmt. to include weekly cadences, reporting, and fielding any ad-hoc requests
- Stakeholder management
- Mentoring and coaching of team members
- Equipment mgmt.
- Project risk and issues tracking and mgmt.
- Quality control of deliverables
- Cross-functional team oversight to include internal and external parties
- Delegation of responsibilities where applicable to project support teams
- Cross-team escalation mgmt.

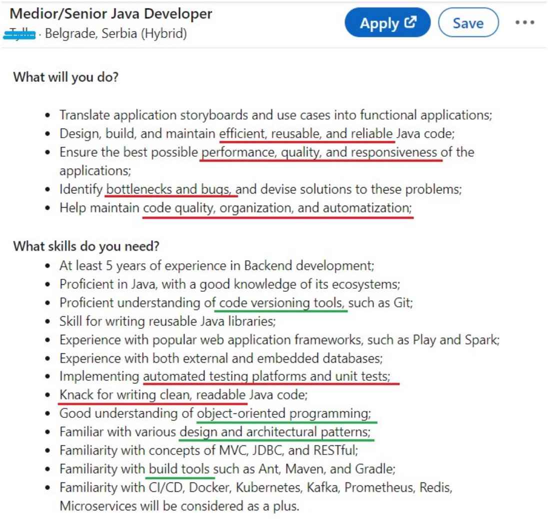
Work Environment

- Team Focus
- Works within a project team environment where results will be based on cooperation with and/or influence of others.
- Projects are normally within the context of a solution team, internal team, or other [redacted] business units.

Success Measures

- Focuses on results and success, conveying a sense of urgency and driving issues to closure
- Meets deadlines and produce high quality work
- Optimizes use of available resources
- Relates to people in an open, friendly, accepting manner
- Shows sincere interest in others and their concerns
- Initiates and develops relationships with others

Figure 4. Example of project manager job advertisement from LinkedIn



Medior/Senior Java Developer

[redacted] · Belgrade, Serbia (Hybrid)

What will you do?

- Translate application storyboards and use cases into functional applications;
- Design, build, and maintain efficient, reusable, and reliable Java code;
- Ensure the best possible performance, quality, and responsiveness of the applications;
- Identify bottlenecks and bugs, and devise solutions to these problems;
- Help maintain code quality, organization, and automatization;

What skills do you need?

- At least 5 years of experience in Backend development;
- Proficient in Java, with a good knowledge of its ecosystems;
- Proficient understanding of code versioning tools, such as Git;
- Skill for writing reusable Java libraries;
- Experience with popular web application frameworks, such as Play and Spark;
- Experience with both external and embedded databases;
- Implementing automated testing platforms and unit tests;
- Knack for writing clean, readable Java code;
- Good understanding of object-oriented programming;
- Familiar with various design and architectural patterns;
- Familiarity with concepts of MVC, JDBC, and RESTful;
- Familiarity with build tools such as Ant, Maven, and Gradle;
- Familiarity with CI/CD, Docker, Kubernetes, Kafka, Prometheus, Redis, Microservices will be considered as a plus.

Figure 5. Example of software development job advertisement at LinkedIn

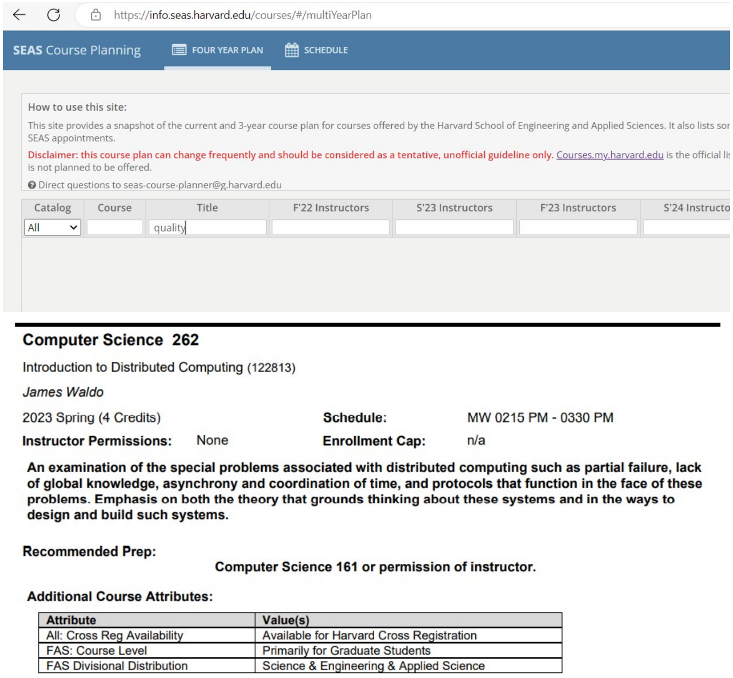
Finally, there is a summary table of software quality aspects in different roles within the software development process, as presented in Table 2.

Project management	**Business/system analyst**	**Software architect**
Quality of project deliverables	Quality of specifications	Appropriate software architecture
Quality of requirements specifications Monitoring of process	Quality of business process analysis and mapping to design	
Software developer	**Software quality lead**	**Software quality engineer** **Tester (manual, automation)**
Clean code	Quality assurance plan	Test cases design
Efficient code	Quality assurance tools, methods	Bug reports
Software product – quality standards	Quality assurance reports	

Table 2. Overview of key software quality aspects addressed in different working roles in software development process

Empirical study on software quality content in university teaching

Aim of this empirical study is to determine the support to software quality-related contents in higher education institutions Study programmes, curriculum as a structure of a study program and each subjects' syllabus were extracted and analyzed from available data from university schools web portals. The data source for the research sample are top-most universities from Shanghai university ranking list [36] and state-owned universities in Serbia.



How to use this site:

This site provides a snapshot of the current and 3-year course plan for courses offered by the Harvard School of Engineering and Applied Sciences. It also lists sor
SEAS appointments.

Disclaimer: this course plan can change frequently and should be considered as a tentative, unofficial guideline only. Courses.my.harvard.edu is the official lis
is not planned to be offered.

Direct questions to seas-course-planner@g.harvard.edu

Catalog	Course	Title	F'22 Instructors	S'23 Instructors	F'23 Instructors	S'24 Instructo
All		quality				

Computer Science 262

Introduction to Distributed Computing (122813)

James Waldo

2023 Spring (4 Credits) **Schedule:** MW 0215 PM - 0330 PM

Instructor Permissions: None **Enrollment Cap:** n/a

An examination of the special problems associated with distributed computing such as partial failure, lack of global knowledge, asynchrony and coordination of time, and protocols that function in the face of these problems. Emphasis on both the theory that grounds thinking about these systems and in the ways to design and build such systems.

Recommended Prep:

Computer Science 161 or permission of instructor.

Additional Course Attributes:

Attribute	Value(s)
All: Cross Reg Availability	Available for Harvard Cross Registration
FAS: Course Level	Primarily for Graduate Students
FAS Divisional Distribution	Science & Engineering & Applied Science

Figure 6. Courses list with search options and catalogue with courses details including teaching content description – example of Harward university

Model for search within educational support include using keywords that represent key software quality aspects – *software project management*, *software architecture*, *system/business analysis*, *software quality assurance*, *software testing*.

Analysis of available data about bachelor and master level study programmes from Top-most universities from Shanghai university ranking list show that there are no study programmes named "Software quality assurance", but mostly there are Computer science bachelor and master programmes. Among courses at computer science bachelor and master programmes there are no software quality assurance as a separate course, but courses within computer science programmes are named after fundamental aspects of computer architecture, programming, software development and modern technologies.

Analysis of study programmes at state-owned universities in Serbia shows that there are no particular study programmes named in the context of "software quality assurance", but within study programmes named "Software engineering" there are courses related to software testing, software development quality, software design patterns, while within study programmes named "information systems development", there are courses related to systems analysis and design, analysis of business processes, project management, software quality management, software design patterns, software testing, software architectures, information security.

The proposed educational framework

This section presents the proposed educational framework to be applied in university teaching, with particular emphasis on quality of software as a product and code quality. The table below presents aspects of software quality – process aspect, functional aspect, non-functional aspect, structural aspect, programming language grammar alignment, quality in use.

QUALITY ASPECT	PROCESS ASPECT	FUNCTIONAL ASPECT	NON-FUNCTIONAL ASPECT	STRUCTURAL ASPECT	GRAMMAR ALIGNMENT OF CODE	QUALITY IN USE
COURSE	Project management	System analysis and design	Human-computer interaction	Software architecture	Program translators/ construction of compilers	Software testing
TOPICS	Software development life cycle Standards Documentation	Business process modeling Mapping design of functions and data to business process elements	User interface Front-end programming	Software modularization, software architectures, design patterns, frameworks, clean code and coding conventions	Lexical, Syntax, semantic aspect, compile time and run time errors	Test based specification, test cases, test automation
METHODS and tools	Agile methodologies	UML	User experience design	Code refactoring	Exception handling	Input data validations

Table 3. Educational framework for teaching software quality at higher education

The proposed educational framework could be used within a single course named "Software quality assurance" or could be divided as topics within different subjects, such as software project management (the process aspect of quality), systems analysis and design(functional aspect of quality), human-computer interaction (nonfunctional software requirements).

Software architecture (structural aspect of quality), program translators/-compilers construction (alignment of code with programming language grammar), software testing (quality in use).

Conclusion

Aim of this paper was to present software quality area from the perspective of research results, industrial needs and educational support.

Limitations of this paper are present in every part of this paper. They are related to the sample formulation and methods that are used for their implementation. Literature review has been based on Google scholar, while systematic literature review would require multiple sources and diversity of search keywords, to have more complete results. Empirical study of job advertisements have been performed upon LinkedIn as a single source of job advertisements text and there is need for more detailed analysis of texts that are included in these job advertisements. Analysis of educational support to software quality has been briefly conducted upon insight (with very much trouble to find data) in web sites of top-most ranked universities at Shanghai university ranking list. Sample of Serbian universities has been selected among state-owned universities, while the complete image of Serbian university space and the position of software quality within the curriculum of study programmes could be obtained with inclusion of private-owned universities as well.

Future research and practical work could be directed towards performing detailed and systematic literature review about software quality. It would be beneficial to include key findings from this paper to educational practice in higher education – to include research results and the proposed educational framework as a source to improve educational practice. New research areas are directed towards creating tools that could automate software development and quality control, particularly in the field of evaluation of coding style and coding conventions (clean code).

References

[1] I. Plaza, R. Igual, C. Medrano, M. Rubio, From Companies to Universities: Application of R&D&I Concepts in Higher Education Teaching, *IEEE Transactions on Education*, Vol. 56, Issue 3, August 2013.

[2] I. Noben, R. Maulana, J. Deinum, W. Hofman, Measuring university teachers' teaching quality: a Rasch modelling approach, *Learning*

Environments Research, 2021, 24, 87–107.

[3] C-Y Chen, P-C Chen, P-Y Chen, Teaching quality in higher education: An introductory review on a process-oriented teaching quality model, *Total Quality Management and Business Excellence*, Vol. 25, Issue 1–2, 2014.

[4] J. Ludewig, R. Reising, Teaching what they need instead of teaching what we like – the new Software Engineering Curriculum at the University of Stuttgart, *Information and Software Technology*, Vol. 40, Issue 4, 15 July 1998, 239–244.

[5] J. Pollack, J. Helm, D. Adler, What is the Iron Triangle, and how has it changed?, *International Journal of Managing Projects in Business*, 11(2), 2018, 527–547.

[6] ISO IEC 25010:2011 standard – `https://www.iso.org/standard/35733.html` (visited: October 1, 2022).

[7] M. Akbar, J. Sang, A. Khan, N. Fazal-E-Amin, Nasrullah, M. Shafiq, S. Hussain, H. Hu, M. Elahi, H. Xiang, Improving the Quality of Software Development Process by Introducing a New Methodology—AZ-Model, *IEEE Access*, Vol. 6, 2018, 4811–4823.

[8] K. Lee, S. Sung Jong Lee, A quantitative software quality evaluation model for the artifacts of component based development, *Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Network*, 23–25 May 2005.

[9] A. Orso, G. Rothermel, Software Testing: A Research Travelogue (2000–2014), Conference FOSE'14, May 31 – June 7,2014, Hyderabad, India.

[10] H. Munir, M. Moayyed, K. Petersen, Considering Rigor and Relevance when Evaluating Test Driven Development: A Systematic Review, *Information and Software Technology*, 2014, `doi:http://dx.doi.org/10.1016/j.infsof.2014.01.002`.

[11] M. Unterkalmsteiner, R. Feldt, T. Gorschek, A taxonomy for requirements engineering and software test alignment, *ACM Transactions on Software Engineering and Methodology (TOSEM)* 23.2 (2014): 1–38.

[12] S. Liu, Validating formal specifications using testing-based specification animation, FormaliSE'16: Proc. of the *4th FME Workshop on Formal Methods in Software Engineering*, May 2016, 29–35, `https:`

//doi.org/10.1145/2897667.2897668.

[13] T. Hilburn, M. Towhidenejad, Software Quality: A Curriculum Postscript, *SIGSCE conference proceedings*, ACM, 2000, 167–171.

[14] C. Burgess, B. Jones, Should software quality be a major issue when teaching first year programming to software engineers?, *Transactions on Information and Communications Technologies*, Vol. 7, 1994, WIT Press, ISSN: 1743-3517.

[15] W. Suryn, Thoughts on Teaching Software Quality Engineering, Proc. of *8th Annual INSPIRE Conference*, 2003.

[16] C. Laporte, A. April, K. Bencherif, Teaching Software quality Assurance in an Undergraduate Software Engineering Program, *Software Quality Professional* 9.3 (2007): 4.

[17] R. Maíra, M. Marques, A. Quispe, S. Ochoa, A systematic mapping study on practical approaches to teaching software engineering, *IEEE Frontiers in Education Conference (FIE) Proceedings*, 22–25 October 2014.

[18] T. Dingsøyr, M. Jaccheri, A. Wang, Teaching Software Process Improvement through a Case Study, January 1999.

[19] K-S Song, Teaching Software Engineering Through Real-Life Projects to Bridge School and Industry, *SIGCSE Bulletin*, Vol. 28, No. 4, December 1996.

[20] A. Neyem, J. Benedetto, A. Charcon, Improving Software Engineering Education through an Empirical Approach: Lessons Learned from Capstone Teaching Experiences, *SIGCSE'14*, March 5–8, 2014, Atlanta, GA, USA.

[21] D. Damian, H. Allyson, B. Al-Ani, Instructional design and assessment strategies for teaching global software development: a framework, Proc. of *28th international conference on Software engineering*, 2006.

[22] D. Rombach, J. Munch, A. Ocampo, W. Humphrey, D. Burton, Teaching disciplined software development, *The Journal of Systems and Software* 81, 2008, 747–763.

[23] I. Richardson, L. Reid, B. Pattinson, Y. Delaney, S. Seidman, Educating Software Engineers for the Future: Software Quality Research through Problem-Based Learning, *24th IEEE-CS Conference on Software Engineering Education and Training (CSEE&T)*, 2011.

[24] B. Sussy, A. Jose, J. Calvo-Manzano, C. Cuevas Gonzalo, S. Tomás,

Teaching Team Software Process in Graduate Courses to Increase Productivity and Improve Software Quality, *IEEE Annual International Computer Software and Applications Conference (COMPSAC)*, 2008.

[25] J. Tuya, J. Garcia-Fanjul, Teaching Requirements Analysis by means of Student Collaboration, *29th ASEE/IEEE Frontiers in Education Conference*, November 10 – 13, 1999, San Juan, Puerto Rico.

[26] S-F Tang, P-L Poon, Teaching Specification-Based Testing, *4th Annual Workshop on Teaching Software Testing*, January 2005.

[27] A. Goswami, G. Walia, *Teaching Software Requirements Inspections to Software Engineering Students through Practical Training and Reflection ASEE Annual Conference & Exposition*, New Orleans, Louisiana, 2016.

[28] B. Clegg, J. Rojas, G. Fraser, Teaching software testing concepts using a mutation testing game, *IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*, 2017.

[29] H. Liu, F. Kuo, T. Chen, Teaching an end-user testing methodology, Proc. of *23 rd International IEEE conference on Software Engineering Education and Training*, CSEE, USA, 9–12 March 2010, 81–88.

[30] N. Truong, P. Roe, Static Analysis of Students' Java Programs, Proc. of the *Sixth Australasian Conference on Computing Education*, Vol. 30, 2004.

[31] R. De Andrade, R. Arakaki, Teaching Software Architecture Quality based on run-time metrics, *International Conference on Engineering Education – ICEE 2007*, September 3 – 7, 2007, Coimbra, Portugal.

[32] P. Gomes, G. Spadon, R. Garcia, D. Eler, Teaching software quality via source code inspection tool, Proc. of *2017 IEEE Frontiers in Education Conference (FIE)*, October 2017, `DOI:10.1109/FIE.2017.8190658`.

[33] A. Alnawaiseh, Y. Helmu, E. Khalil, A New Software Quality Model for Academic Information System "Case Study E-learnimg system", *International Journal of Scientific & Technology Research*, February 2020.

[34] O. Gotel, C. Scharff, A. Wildenberg, Teaching Software Quality Assurance by Encouraging Student Contribution to an Open Source Web-Based System for the assessment of Programming Assignments, Proc. of *13th Annual Conference on Innovation and Technology in Com-*

puter Science Education (ITiCSE 2008), Madrid, Spain, June 30 – July 2, 2022.

[35] N. Suradi, S. Kahar, N. Akmal Jamaluddin, Identification of Software Quality Characteristics on Academic Application In Higher Education Institution (HEI), *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 2–7 October, 2018, 133–136.

[36] Shanghai university ranking list, `https://www.shanghairanking.com/rankings/arwu/2022` (visited: October 3, 2022).

Ljubica Kazi*
University of Novi Sad,
Technical Faculty "Mihajlo Pupin",
Djure Djakovica bb, 23000 Zrenjanin, Serbia
Corresponding author: `ljubica.kazi@tfzr.rs`