UNIVERSITY "POLITEHNICA" of BUCHAREST

DOCTORAL SCHOOL ETTI-B

Decision 493 **of** 06.02.2020

SUMMARY OF PhD THESIS

RESEARCH ON VERIFICATION AND VALIDATION TECHNIQUES FOR INFORMATION SYSTEMS

PhD Candidate: **Eng. Sabina-Daniela AXINTE**

DOCTORAL COMMITTEE

Chair of the committee	**Gheorghe BREZEANU, PhD**	from	**University POLITEHNICA of Bucharest**
Doctoral advisor	**Ioan BACIVAROV, PhD**	from	**University POLITEHNICA of Bucharest**
Reviewer	**Mircea POPA, PhD**	from	**University Politehnica of Timișoara**
Reviewer	**Daniela-Elena POPESCU, PhD**	from	**University of Oradea**
Reviewer	**Paul ȘCHIOPU, PhD**	from	**University POLITEHNICA of Bucharest**

BUCHAREST 2020

Table of Contents

Chapter 1. Introduction

An ever-increasing number of devices across all industries are operating through computer software, and we live in a constantly shifting world from the perspective of trust afforded to said software. This makes the proper and safe operation of virtually any piece of machinery or equipment a real-world necessity.

Quality is a vast, industry-focused field, be it economics, engineering or computer science. The term "quality" primarily refers to a product's conformity with regards to its purpose, market focus and customer demands, regardless of its type. Multiple definitions and visions of quality exist, but its first definition was put forth by Walter A. Shewhart in the early 20th century: "There are two common aspects of quality: one denotes it as objective reality, independent of the existence of man. The other refers to what we think, believe or feel as a result of this objective reality. In other words, there is a subjective side to quality." [1]

In software engineering, quality pertains to two standalone, yet interconnected concepts. Firstly, it defines the *functional quality* of software, quantifying the degree of likeness between the final product and the initial specifications, purpose and requirements. It represents a measure of a system's correct design. The second concept is the *structural quality* of software and relates to the non-functional requirements met by the product, which support the delivery of functional requirements, like robustness and maintainability. It reflects the measure of a software's proper operation.

Software testing is an investigation with the main goal of gathering information on the quality of the product or system under test. It provides an independent, objective outcome which aids in the decision-making process involving risks innate to software development.

The growing diversity of the online world generated a significant increase in the demand for complex Web-based applications, with ever stricter criteria for reliability, usability, security and interoperability. Due to customer pressure, Web application testing is often ranked lower on a project's priority list, since it requires a considerable amount of resources, which impacts the quality of the final product. Because of the number of different technologies employed in the development of a Web application and the combination of a large number of component types (new, inherited, generated from third-party libraries, media, etc.), and the heterogeneous nature of production environments (various types of platforms, Web servers and browsers), the potential number of globally-situated users, and the concurrency of their access, Web application testing is, generally speaking, far more difficult than traditional software testing.

1.1. The field of the doctoral thesis

Quality assurance in software engineering is of great importance to the modern world and constant evolution is forced upon it by the ongoing, accelerated development efforts worldwide, and by the multitude of aspects which alter and transform the way software products are developed and used. We will present essential fragments of field

nomenclature, in order to establish a common language, as well as a collection of the most significant software flaws which occurred throughout history. These are detailed in descending order of human and financial impact, as opposed to chronologically, starting with the first recorded use of the term "bug" [30]. A summary of the field's metamorphosis and current trends are also presented.

1.2. The goal of the doctoral thesis

Given the field's complex evolution and aspects, the main goal of this paper is to ***offer solutions for analyzing and optimizing the painstaking processes which form the basis of computer software quality assurance***. At the same time, various ways to improve upon the final software through increasing the efficiency of each stage of the development lifecycle will be underlined.

Constant learning throughout one's lifetime is indispensable to mankind, particularly in the IT&C industry. Nevertheless, a limited number of universities and learning centers are offering courses on the subject of quality assurance. Therefore, we aim to design an e-learning platform with the stated goal of preparing students in the chosen field and providing a practical example of quality-centric software development.

In order for these main objectives to be accomplished and developed based on coherent, detailed research, we propose the following adjacent goals:

- Performing an analysis of the current educational offering, both national and online;
- Completing a QFD analysis by virtue of the House of Quality as a means of translating user requests into technical specifications for the design phase of our e-learning platform;
- Studying software development methodologies and processes in great detail and in accordance with the various product archetypes;
- Designing a transitional diagram, as well as a fault tree, in order to illustrate the reliability computation of a software product;
- Optimizing testing processes and delivering a best practice guide on the design and compilation of documentation pertaining to the verification and validation processes;
- Proposing solutions for maximizing the reliability of the final product as early as the planning phase, regarding both its functional and non-functional requirements;
- Developing the aforementioned platform in accordance with the needs of a software quality assurance engineer;
- Representing practical testing methods and techniques across all verification and validation levels;
- Identifying and validating auxiliary tools fit for testing various product characteristics, as well as automating certain manual labor.

1.3. The content of the doctoral thesis

In the introductory **Chapter 1**, we defined the key domain terms in order to establish a common language, we highlighted a history of events which formed the basis of fundamental development in the software quality industry and we presented the most historically significant software flaws, enumerated according to their impact. We also analyzed the current state of the software quality assurance field and specified the goal of our paper.

Chapter 2 encompasses the initial stages of software development: project initiation and identification of a favorable development model. We performed a quasi-exhaustive analysis of existing information providers for software testing, from both the academic and private fields, which revealed their particular weaknesses. Subsequently, we showed an objective alternative to both Agile and traditional approaches and concluded with a comparative analysis of them. Furthermore, we produced an optimization proposal of the development process by virtue of the project team's composition.

Chapter 3 incorporates means of qualitative product synthesis with the help of a Gantt chart based on the QFD analysis, the field's directives and security standards, as well as the prescribed methodology for structuring testing endeavors. Additionally, we elaborated suggestions for organizing a test plan which reflects the techniques adapted to the product we wish to test. Finally, we expanded upon the prospects of security, usability and reliability improvements of a product in its design phase and completed the analysis of the architectural environment.

Chapter 4 addresses the implementation and verification stages of the software development lifecycle. Firstly, we introduce the taxonomy of software defects and specific methods of prevention. Secondly, we outline the steps taken to develop the e-learning software, pursuant to the study of the quality assurance engineer's profile. Recommendations were made regarding the execution of test scenarios pertaining to the functional testing of the aforementioned system and assurances were obtained with respect to the product's conformance to the initial specifications.

During **Chapter 5** we undertake the issue of non-functional requirement validation of computer systems. We present ways to establish the security level of a software product, exemplified for the designed e-learning application. Furthermore, we set forth processes for boosting the efficiency of usability testing through auxiliary tools.

The doctoral thesis is finalized with a chapter for **Conclusions**, which outlines a synthesis of concepts expounded upon throughout this paper, its findings and our contributions. The chapter also contains a list of works published throughout the doctoral research cycle and suggestions for further advancing this research.

The final portion of the thesis is comprised of **four annexes** with additional data which supplemented the thesis research and helped achieve its goals.

The paper also contains separate sections for the *List of tables*, the *List of diagrams*, and the *List of abbreviations*, as well as the *Bibliography*, to support in obtaining its findings and structure the presented work.

Chapter 2. Applied analysis techniques for managing complex software solutions

This chapter is comprised of aspects pertaining to project initiation for the development of a computer system. During this phase we identify the issues it must solve, the type of users we design for, the feasibility of our proposed solution, as well as our main goals.

2.1. Project initiation

As we set out to analyze the educational opportunities available at the national academic level, in section 2.1.1 we initially studied data provided by the Ministry of Education and Research on 99 public and private universities [41] and found the number of people who sign up for IT&C curricula is steadily decreasing year over year. We followed up by investigating in extenso the 17 faculties across 12 university centers that we selected based on choice relevance criteria in the field. We found 15 courses which deal with the subject of software quality, as well as 16 courses which touch upon this subject by dealing with one of its branches. Eventually, we identified 23 Master's programmes which were determined to have significant shortcomings based on their curriculum and study charts.

Consequently, the following two subchapters include scrutiny of online material available as an educational offering. We focused on ISTQB, the organization with the most prestigious certification in this field, as well as several online platforms. Using a SWOT analysis, we concluded that there is no flexible solution which presents information in a correct, coherent manner, able to satisfy the needs of people who desire to be or are already professionally active in this field.

To determine the viability of an adjacent solution, we performed a market study based on a survey distributed to groups of industry specialists via social media platforms, both locally and internationally. The results breakdown emphasized that there is a need for a platform which alleviates the issues with the aforementioned offerings and allowed us to identify the project's main objectives.

We followed up by translating user requests into technical specifications by means of a QFD analysis (Fig. 2.19), with illustrative support from the *House of Quality* [50]. After assessing dependencies and impact, we reflected that the functionalities which must be planned and implemented are: a clear hierarchy of accurate, concise information, verification tests, presentations of testing and automation tooling, as well as high-quality technical content. We also surmised that these should be offered in a low-latency, high availability manner, backed by great compatibility with all types of user devices.

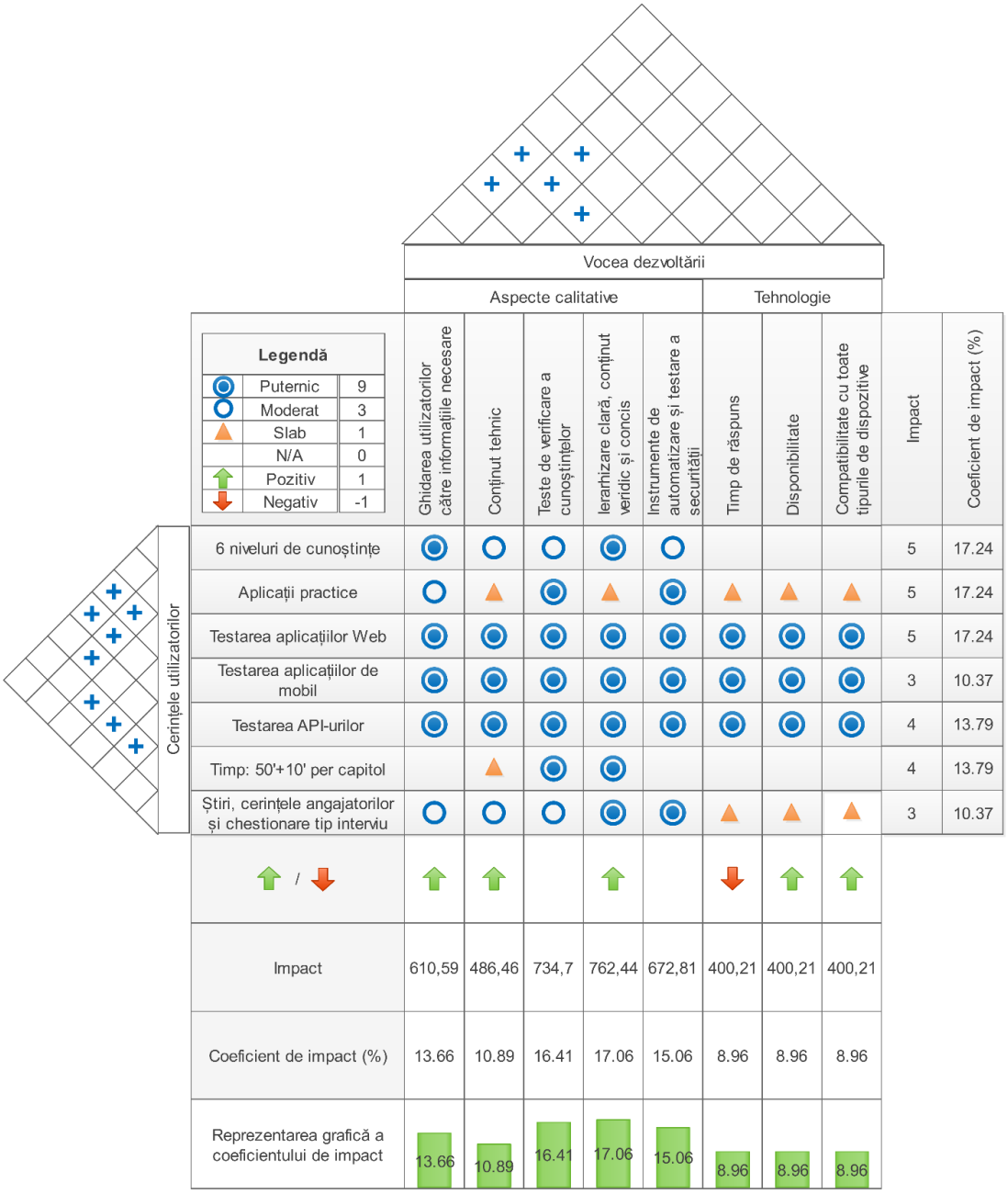


Fig. 2.19 *House of Quality applied on user-expressed requests*

2.2. Planning methodologies for a software project

Since history has proven that correctly choosing a software project's development methodology is the key to its success, this section entails the analysis and determination of our optimal methodology. We expounded upon the Agile approach [52], as well as traditional ones [53] employed during software development lifecycles. Subsequently, we performed a comparative analysis between them in the interest of determining the ideal one for a medium-sized project and decided upon the Iterative Incremental model (IID). To improve upon it, we employed a SWOT analysis (Fig. 2.27) which allowed us to determine potential weaknesses.

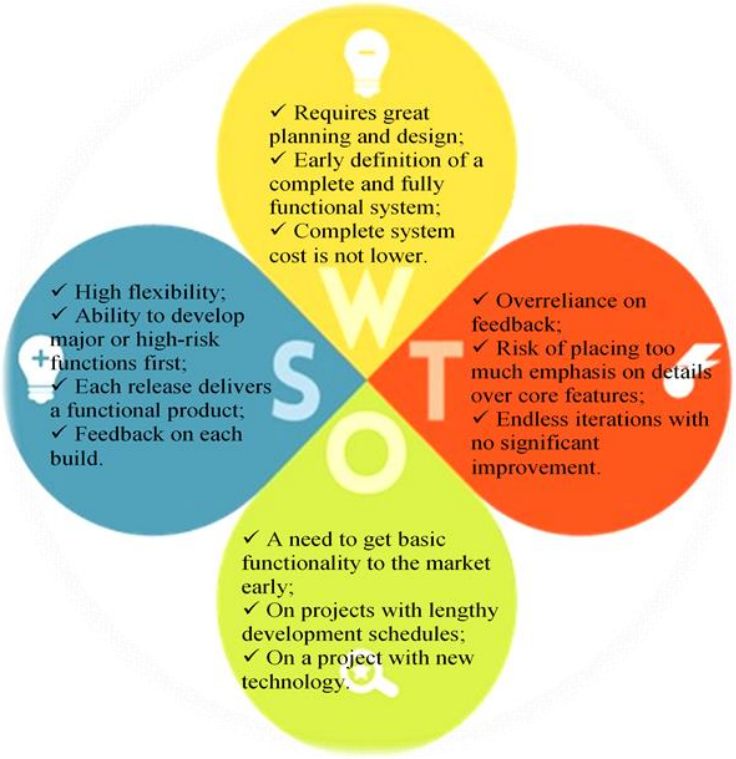


Fig. 2.27 *SWOT analysis of the Iterative Incremental model [55]*

Thus, we established our proposal of the optimization solution (Fig. 2.28) in the context of a medium-sized software product, with the project running for a predetermined amount of time. Improvements refer to the manner of planning [57], the structure and collaboration of the development team, as well as required procedures for continuous enhancement of the development process.

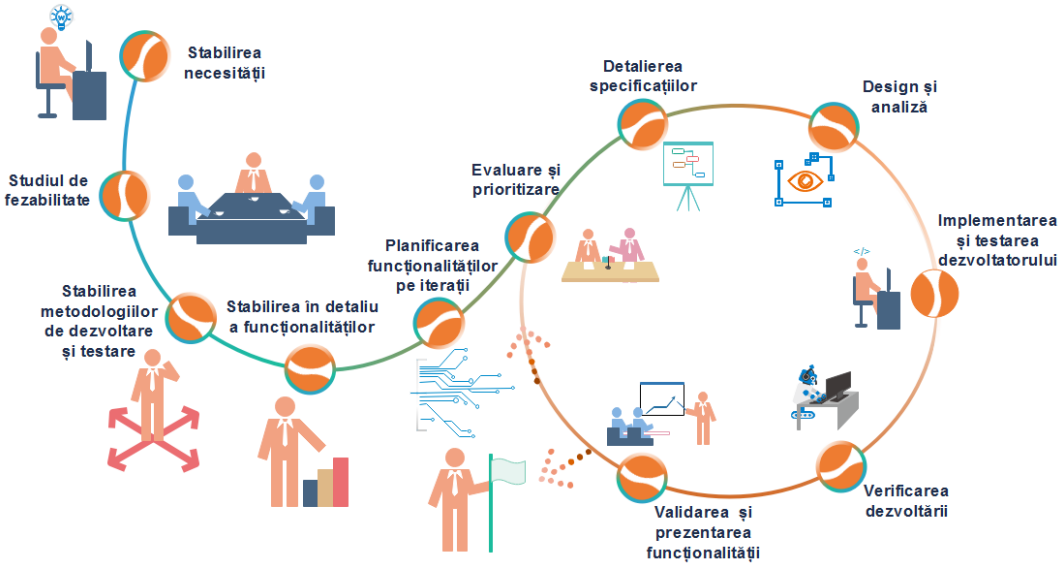


Fig. 2.28 *Proposal for the software development model*

Finally, we supply a model for a sustainable team engaged in a complex, dynamic environment, able to ensure high reliability product development in a comparatively short timeframe [59].

Chapter 3. Studies on the design and implementation of a software product

Chapter 3 aims to focus on the planning and design stages of a computer system's development lifecycle.

3.1. Planning the product's correct development

To achieve this, we began with the construction of the transition state diagram (Fig. 3.1), based on the QFD analysis' conclusion of segmenting and arranging the content into a hierarchy defined based on the user's current level of knowledge. We deemed it necessary to implement three distinct types of users with clearly delimited roles: administrator, instructor and student.

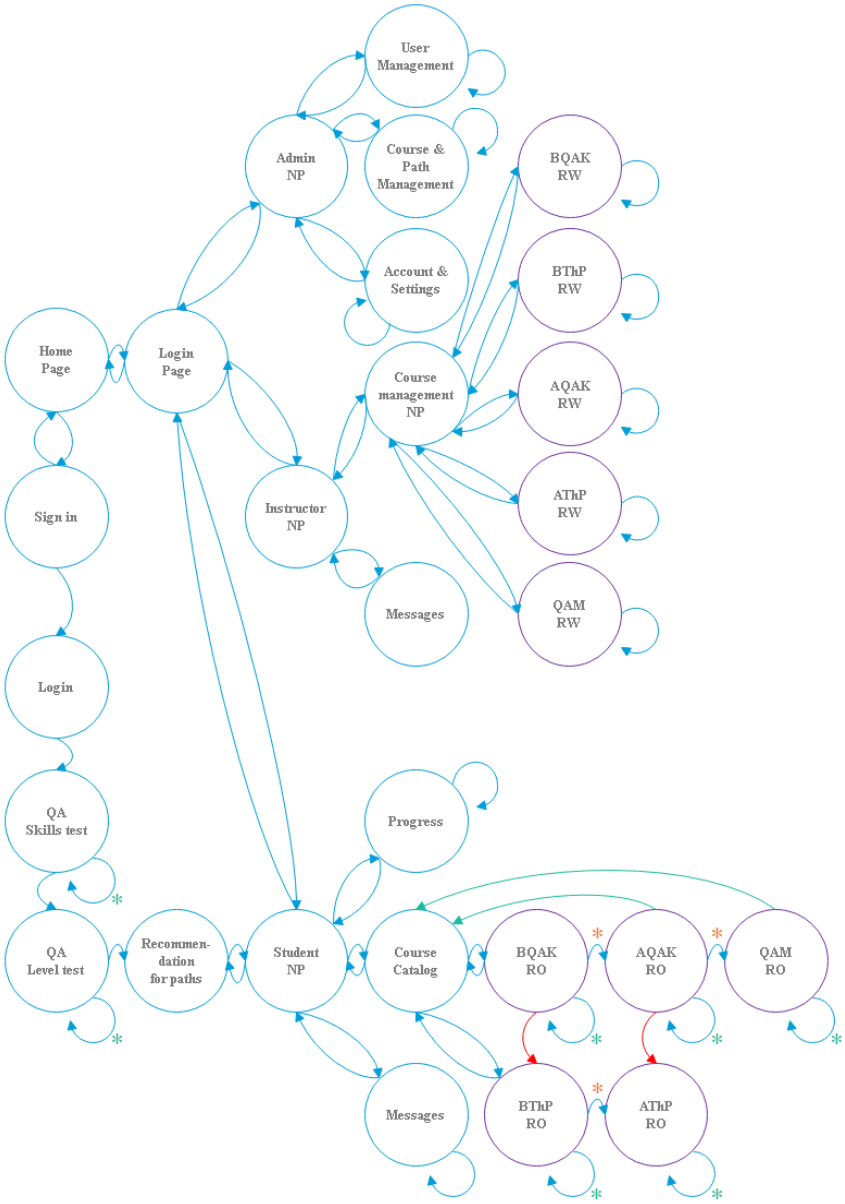


Fig. 3.1 *The transition state diagram of the e-learning application*

We continued by producing a Gantt chart for an illustrated representation of the planned activities and their timeline. Thus, we set the project initiation date and outlined four main stages: planning, design and implementation, verification and validation. In the implementation stage we allocated distinct resources based on the level of research required to create the content for each module.

Subsequently, we analyzed the Web product security directives in order to proactively mesh them into the design stage of the Web application. We included ISO standard 27002 [62], which involves the practical guide for proper information security management and the information characteristics, as well as ISO standard 27001 [62], detailing the security techniques applicable doing the PDCA cycle (Plan-Do-Check-Act). Likewise, we integrated the EU GDPR directive [65] (Fig. 3.4), which facilitates classifying personal information based on its sensitivity, and regulates obtaining, manipulating and storing it.

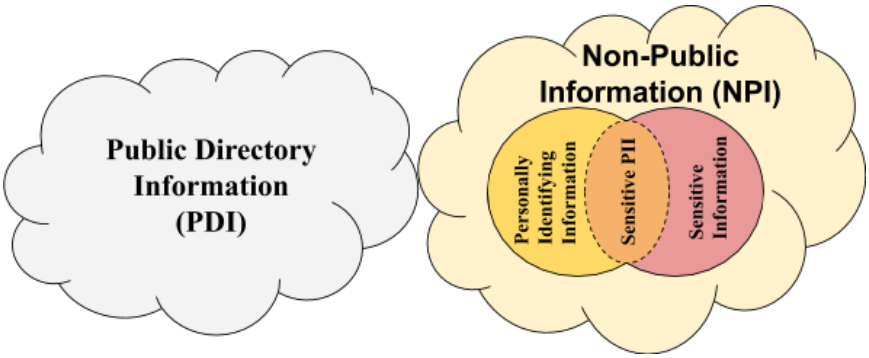


Fig. 3.4 *Data classification per the GDPR [66]*

Additionally, we considered a series of best practice guides [69][70] which help solidify the foundation of processes for the secure manipulation of handled data. We detailed their application during the chosen project's development.

3.2. Establishing the methodology and the required documentation for structuring the testing activities

During this section, we approached the methodology specific to testing activities, as well as the accompanying documentation required for its proper progress. Following a detailed study of testing techniques [73][75], we underlined the fact that static ones must never replace dynamic ones, due to them being complementary and thus used as such, because they tend to efficiently uncover different types of faults. Identifying potential bugs in these early stages leads to much lower costs and efforts to revise and improve product quality. Also, we presented the testing levels during which quality assurance mediation is necessary [8]. We concluded with a comparative analysis of testing techniques in accordance with their associated level and the proposal for optimization (Fig. 3.10) would result in an optimal level of reliability by interpolating the two testing approaches.

Fig. 3.10 *Testing optimization by interpolation of the two techniques*

In order to ensure proper documentation of verification and validation activities we expanded upon the content required for a proper testing plan [75][76] while also explaining the required notions. Finally, we devised a series of practical guidelines for establishing such a test plan which cover the team's involvement in the development activities, the process of redacting test scenarios and cases, their prioritization based on impact, and periodic reviews which ensure improvement and coverage of all vital functionality that is added in the future. These were applied by producing a test plan for our product, which can be found in synthesized form in Annex 3.

3.3. System design

The platform's design was handled next, with a focus on error prevention [77]. Firstly, we presented an evolution of expenses caused by fault remediation based on the stage in which the fault is introduced [78][82]. We outlined proposals for improving the non-functional security characteristics [89], as well as usability, through proper implementation. For the former, we documented ways to employ secure servers which offer technical support for any software products, their strictness being directly proportional to their field of business and project goals. We set forth solutions for ensuring and protecting application access based on well-defined roles and privileges, as well as methods of preventing privilege escalation attacks meant to reach information outside the current user's scope. In order to optimize usability, we focused on a unified platform design, fluid navigation when taking actions throughout the application, error management and user notifications, as well as the language, formatting and graphic design of the actual content [93].

Since we desire total system reliability to be as high as possible, we conducted an FTA (Fault Tree Analysis) [95] in order to improve it, should it not be up to par. Its top event was chosen to be an inability to function properly or limited access to the e-learning platform. Potential causes were delimited into two different sources, end user and computer system. We detailed the base events for each of them according to their provenance, software (Fig. 3.13) and hardware (Fig. 3.14). We set base event probabilities according to hardware component error rates [94][98][99], to results we obtained after constructing a prototype, as well as our experience in working with complex computer systems.

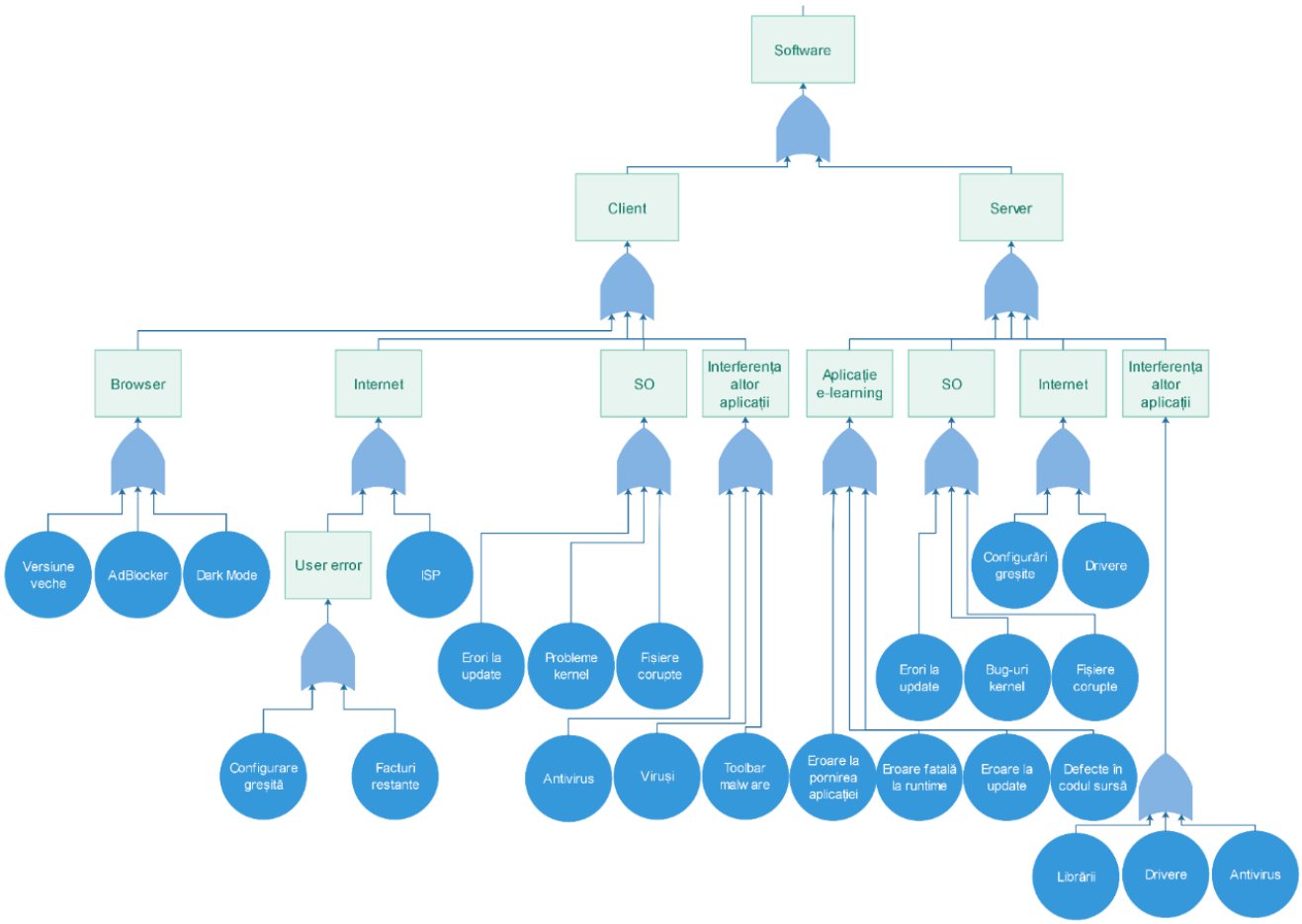


Fig. 3.13 *Software Fault Sub-tree*

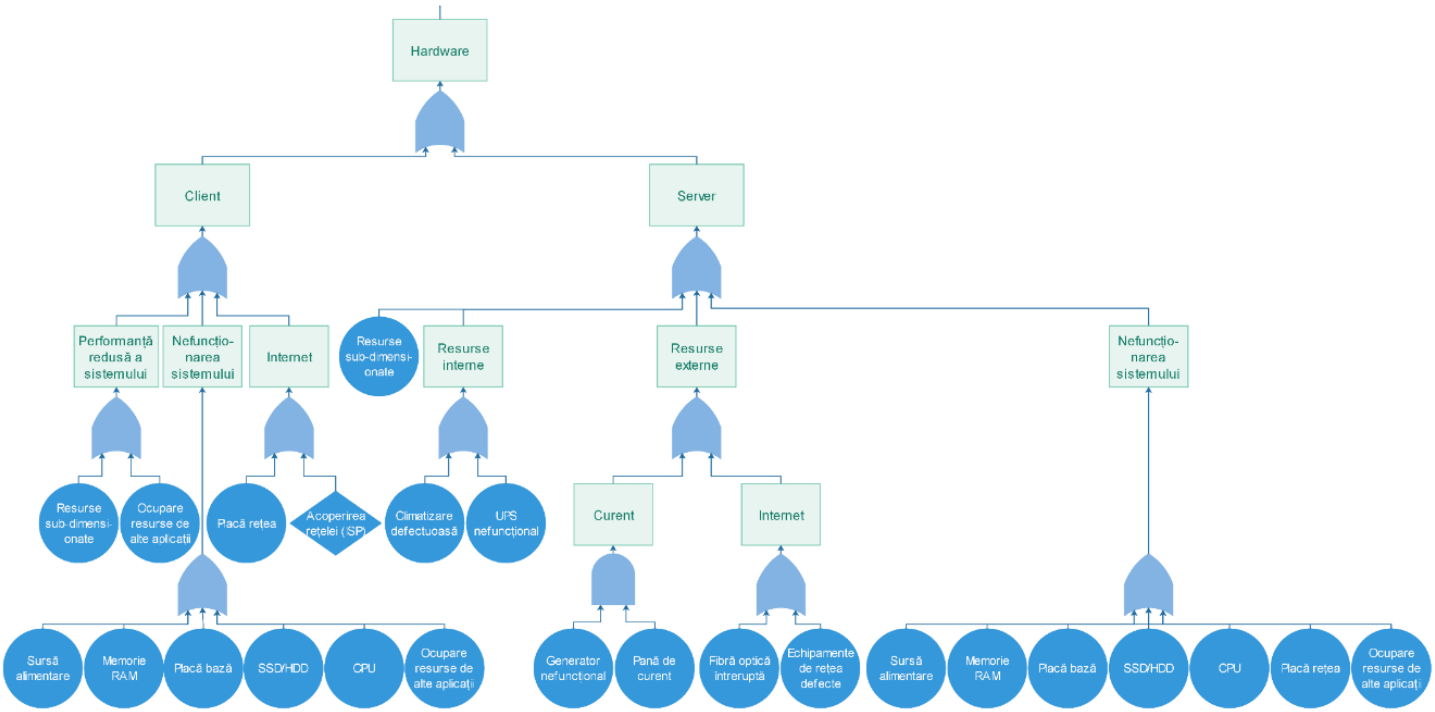
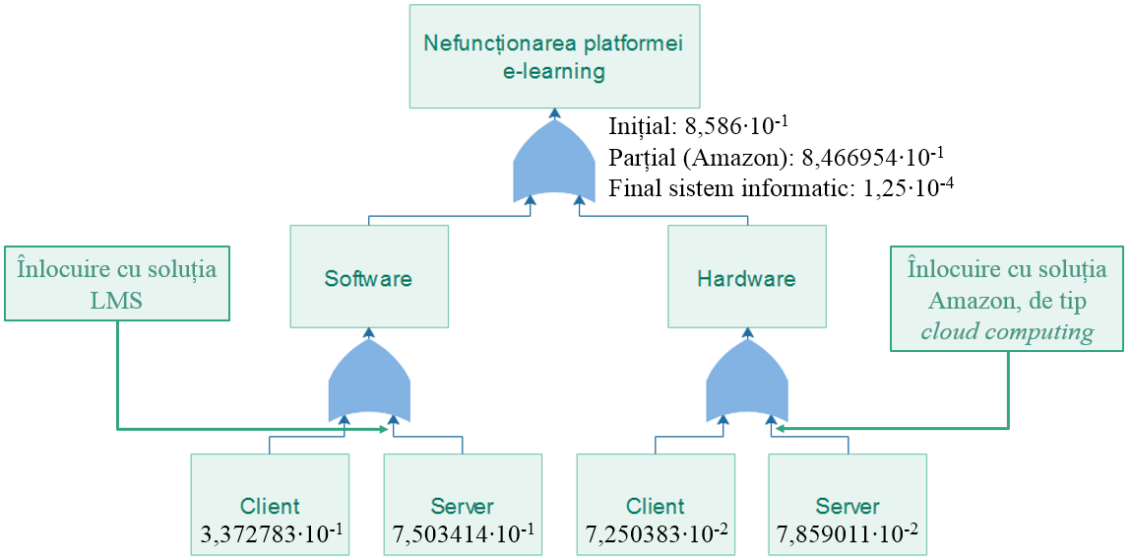


Fig. 3.14 *Hardware Fault Sub-tree*

Finally, we computed the top event probability and found two potential directions for improvement, on the hardware and software branches. In order to diminish the potential number of downtime days from 28.7/year, we adhered to a *cloud computing* solution. After an analysis of the existing options, we selected Amazon's cloud platform as the most reliable one, lowering the amount of downtime by 14.61 days/year [101]. Because the reliability of our software solution did not match our expectations, we opted for a *Learning Management System* solution for which we also analyzed our options. The conclusion showed TalentLMS as ideal for both the functionalities in the project goals, as well as the security and usability demands mentioned previously. To validate this decision, we analyzed the integrated services, solution infrastructure [107] and security policies [112][113] of the architectural environment and concluded that the new proposal for our system architecture complies with all directives of the design and implementation stages and that the new value for the probability of the top event in the FTA, excluding client's components (hardware and software), is $1{,}25 \cdot 10^{-4}$, resulting in a proper operation rate of 99,9874%.



Augmented FTA diagram

Chapter 4. Functional verification through product implementation testing

This chapter is dedicated to the implementation and functional verification of the desired product. We analyzed the fault taxonomy based on its introductory stage and proposed prevention methods for each of them. We also presented the activities associated with product development.

4.2. Application development

During the initiation stage we chose the subdomain name so as to make it representative and easy to remember, created a logo and a favicon according to the application's purpose and selected the necessary settings for language, time zone and date format. We also created a personalized welcome page, activated SSL encryption for our subdomain, enforced strong password usage and restricted concurrent access for a single user. We then built a privacy policy page (Fig. 4.3) in accordance with the GDPR regulation [118].

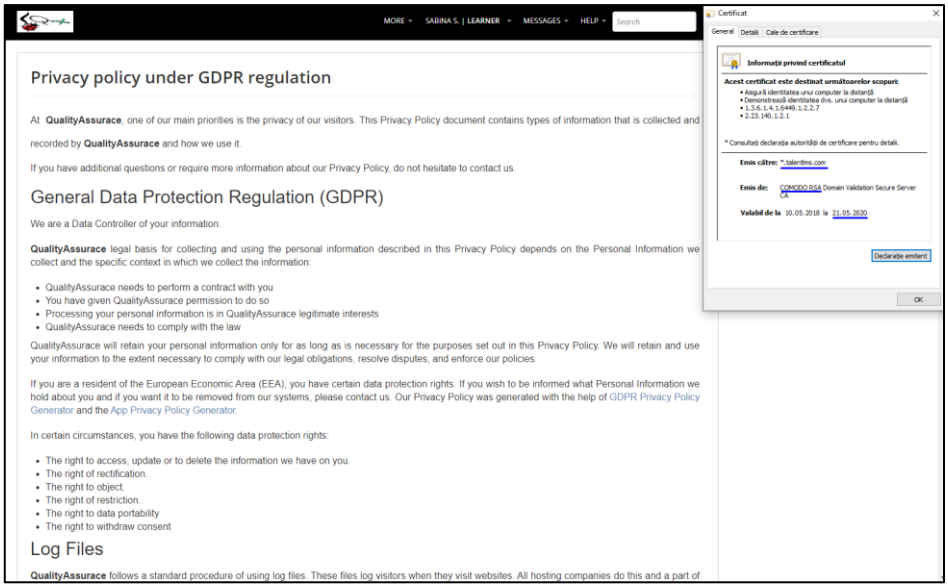


Fig. 4.3 *Application – Privacy policy page as per the GDPR*

We also created custom pages for the presentation of our product's goal, means of contact, as well as the standard HTTP 404 response (page not found). We proceeded to outline the uniform user interface composition (Fig. 4.10), which will be used by all modules developed later. To do so, we selected a color palette which stimulates attention and creativity, as well as a series of fonts and styles which subtly impose focus on key elements within a page.

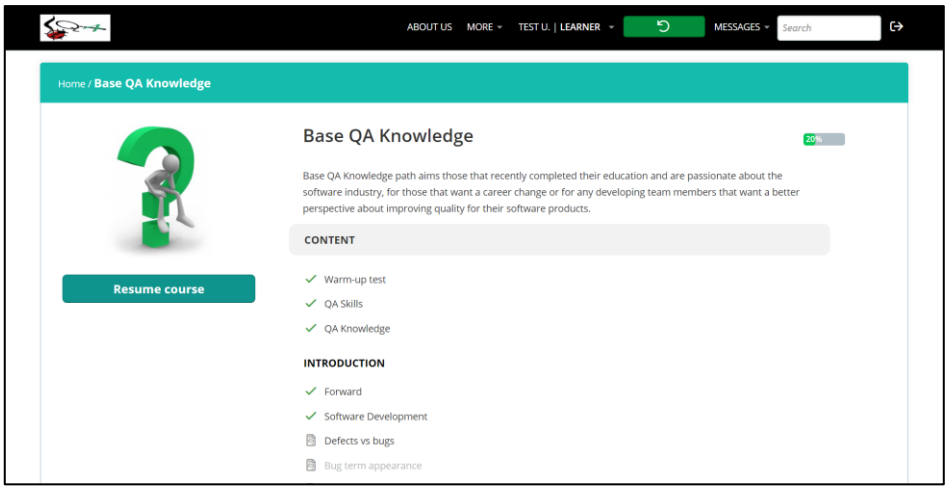


Fig. 4.10 *Theme application within the Base QA Knowledge module*

Finally, we created the module based on the analysis of a software testing engineer's profile, because they are the key product demographic. We outlined intrinsic characteristics which are mandatory for daily activities within the software development environment and created an aptitude determination test based on them, which can be found in Annex 4.

4.3. General software testing principles

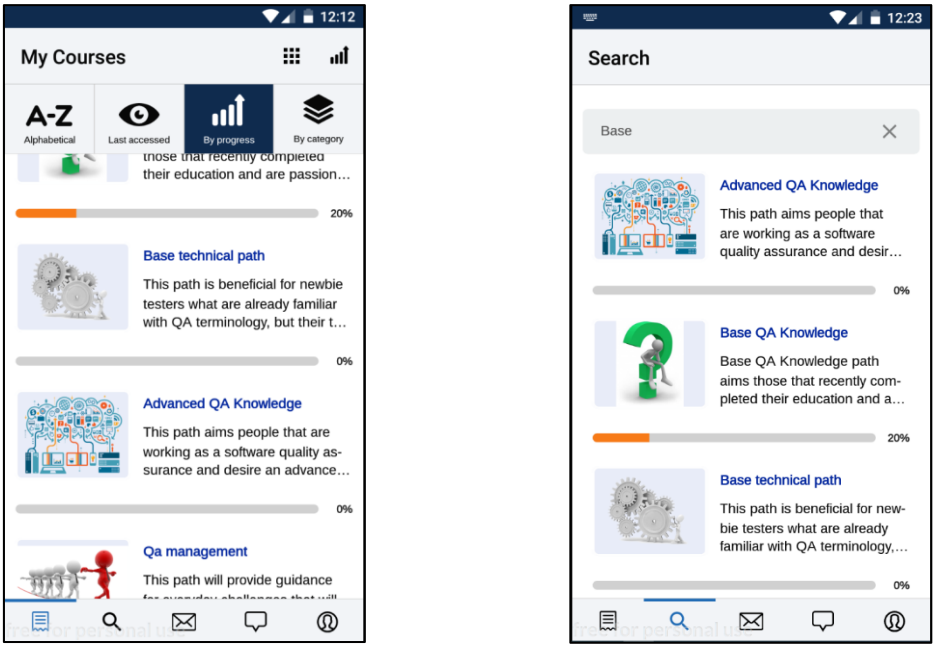
The success of the testing process relies on a series of methodologies and procedures, as well as a set of axioms (principles) developed over time through a variety of means and sources which serve to prevent and eradicate faults, compiled by Ron Patton in "*Software Testing*", one of the reference works of the field [122]. Among the chief aspects we mention that testing should be started alongside the project initiation, as well as the inability to perform exhaustive testing due to considerable delays in product releases, which can also lead to an infinite number of test scenarios for highly complex systems.

4.4. The empirical approach to computer system testing

We propose an empirical approach to initiating verification while creating and following the test plan, inferred from personal experience and the exploratory testing technique. For the latter, our recommendation is to involve a specialist that is unaffiliated with the project, who will follow the logical flow of actions within the application as they complete business and testing flows, and who must persist in areas susceptible to bugs. Good organization is mandatory when detailing the steps to reproduce a particular case, as well as when following up on the evolution of an issue's resolution.

4.5. Functional testing of a software system

We continued by presenting the functional tests [8] covering the previously mentioned specifications in accordance with our completed test plan. The initial testing scenarios focused on ensuring crucial requirements such as the ability to access the main page, the ability to create and log in with an account, as well as the proper display of the navigation page according to the account's associated role. We then devised a test suite for verifying vital functionality in depth, such as selecting a category for more thorough study, viewing modules and courses and the ability to use the messaging system. Finally, we added a series of scenarios for checking the level of product interoperability with the e-learning platform's mobile application (Fig. 4.18).



(a) (b)

Fig. 4.18 *Results obtained after case test 10*

We tracked the ability to connect, the accuracy of user data, the level of information access, proper passage through modules, as well as mobile-specific functionality like downloading content for later viewing without an internet connection, accessing Web versions of courses and cache invalidation. After testing, we performed the necessary fixes and marked the system as conforming to specifications.

Chapter 5. Validating the computer system

An organization must ensure that resources for risk management and treatment are always available to address new and evolving threats and vulnerabilities and to properly keep the management team informed. To that end, we approached validation of non-functional specifications in this chapter, with a focus on the topic of security in the first part.

5.1. Security probing based on the risk analysis

We initiated the system's risk analysis with the intent to probe its defenses and weaknesses from a security point of view. Firstly, we analyzed potential vulnerabilities to a system, such as physical damage, natural phenomena, key service breakdowns, functionality and data compromise, as well as unauthorized actions [134]. Secondly, we analyzed local network or internet-based attacks and the potential reasons for software vulnerability [130]. Thirdly, we presented a classification of attackers based on their intent, level of knowledge and impact. Furthermore, we highlighted the risk treatment stage of the associated management process, focusing on treatment through reduction, acceptance, avoidance or transfer, based on the risk context. We explained why risk evaluation must be performed periodically, in order to handle changes which may influence the result of the evaluation and to ensure risk reduction to acceptable levels. We concluded our analysis by identifying that comparatively higher risk as it pertains to our system is represented by programming errors amounting to a failure to validate and sanitize user requests and input.

Subsequently, we performed actual security testing for our previously verified product. To do so, we focused our validation scenarios on areas with a high vulnerability quotient to input sanitization programming errors. Scenarios were segmented into two complementary sections, manual and automated, to ensure a multi-level security perspective. Both testing methods have their strengths and drawbacks and should be utilized based on the type of product, the areas or functionalities under test, volume, engineer training and qualifications, available time and resources. Testing process automation is desirable in a multitude of cases and can generate significant benefits when the return on investment is considerable and resources allow it.

During manual testing we considered ways of performing SQL injection on the login form, as well as from the authenticated user context, in order to gain information beyond a user's current privileges. Afterward, following a login, we performed URL manipulation testing. We also attempted to engineer XSS (Cross-Site Scripting) attacks [135] for the login form and authenticated search fields. No major vulnerabilities were detected during manual testing.

To facilitate automated testing, we called upon the Vega and ZAP software tools [138], which were designed for security validation prior to end user releases. These identify vulnerabilities through scanning and attempting to induce known problems from a predefined list, most of which are covered by OWASP, the organization with the most significant impact in security attack prevention. Following the analyses generated by

these tools, we found no critical or major vulnerabilities. However, we did uncover a series of notices and warnings of limited impact potential, such as the absence of anti-CSRF (Cross-Site Request Forgery) tokens, or missing content-type headers, as well as some improper cookie settings. The latter is the most notable finding, missing security flags on 40 different cookies.

Based on the security test results, as well as the risk and architectural environment analyses, we declared the computer system as secure, both physically, due to the hardware being in a secure location with deployed systems to aid in risk mitigation, as well as software-related, given the state of the key services (API and database) and the absence of any critical or major vulnerabilities.

5.2. Usability testing

During usability testing we set out to ensure that the product is fit for purpose. To that end, we proposed a method of optimizing the efficiency of responsiveness and compatibility testing that was devised as a result of the analysis of worldwide mobile device traffic [143]. Accordingly, we determined the browser types and versions in use and selected seven of the most relevant ones. We followed up by correlating with their availability on different operating system versions and produced twelve test suites which are able to cover the vast majority of mobile users. In order to augment the relevance of these tests, we chose a number of six mobile device models which encompass the most commonly used screen resolutions and match the aforementioned browsers and operating systems. Subsequently, we proposed three auxiliary tools to enhance this type of testing [92]. The Chrome browser's responsiveness module is recommended for testing during development, in order to identify potential flaws at an early stage, but for the final validation the use of an application which offers virtual mobile devices, Genymotion, due to its resource virtualization and internet connection modulation capabilities producing results with real-world relevance. We also selected the Uptime tool [147], which can validate the loading speed of content based on resource type, as well as application responsiveness in its entirety. Following tests using these tools we identified a series of image formatting issues, as well as high loading times for picture resources, and were able to resolve them.

Chapter 6. Conclusions

The final chapter presents the main aspects covered by the doctoral thesis and synthesizes both its results and the original contributions. It also contains the list of works devised and published during the doctoral internship and highlights potential directions of development for the current research.

6.1. Obtained results

This section presents the results obtained throughout the doctoral internship and details them according to this paper's chapters.

In **Chapter 2** we follow the initial stages of software product development: identifying the addressed issues, the target demographic and user typology, the feasibility of our proposed solution, the main objectives and the optimal development model.

We performed a quasi-exhaustive analysis of current software testing information providers, both within the academic realm and the online environment, highlighting their specific drawbacks. This in extenso analysis was performed on 17 faculties across 12 university centers selected based on relevant field criteria. The outcome of this research was that the diversity of available courses and the applicability of the presented theoretical notions suffer from significant deficiencies. Subsequently, we studied the available online alternatives and presented their advantages and flaws compared to the academic offerings through a SWOT analysis. This solidified the demand for an adjacent solution which outclasses existing options through flexibility and content quality.

We performed a market study through a survey answered by 374 field specialists and aspirants. After gathering and dissecting the results, we found this to be an ongoing issue with a real, feasible solution, and we proceeded to categorize participant demands. We built the House of Quality as an illustrative support for our QFD analysis, in which we proposed a series of functionalities for later development. According to the previously selected user requests, we outlined the scope and objectives of this e-learning platform.

The chapter's second section lays out the optimal development methodology selection for our desired project. Firstly, we presented both Agile and traditional approaches, and performed a comparative exposition. The logical conclusion was that, for a medium-sized software project with a fixed timeframe, the most viable option would be the Iterative Incremental model. In order to validate this conclusion, we performed a SWOT analysis which identified potential areas of improvement in the model's activity planning and user acceptance testing procedures. To alleviate these deficiencies, we elaborated an IDD model improvement proposal which specifies a 25-30% allocation of the project's total runtime for the planning phase, the creation of a dedicated, autonomous, multidisciplinary team with a focus on collaboration, a finer granularity of complex functionality, as well as the implementation of procedures necessary for software development. Additionally, we devised a proposal for a sustainable project team, from the perspective of its constituency, capable of developing the Web interface, the Web service structure and the database transactions in an optimal timeframe, assisted by

solutions provided by a software architect and supported by a business analyst. We also present the contributions of quality assurance engineers in yielding a reliable product, which obeys its functional specifications and meets the demands of its end users.

In **Chapter 3** we present means of qualitative product building by utilizing security directives and standards in the field and the required methodology for structuring the testing activities. Firstly, we develop the state transition diagram which encompasses access privileges throughout the application from three distinct user type perspectives (student, instructor and administrator), as well as the functional aspects established following the QFD analysis. We then construct the Gantt chart to better perceive the required effort for solution development. In order to facilitate a high degree of security in our product, we analyze ISO standards (ISO 27002, ISO 27001), the EU legislation (GDPR) and best practice guides and define the essential product characteristics to be applied in subsequent stages.

We additionally construe suggestions for developing a test plan based on a technique tailored to the product under test, pursuant to which we establish the verification and validation plan within our project. We also highlight methods to improve the security and usability of the final product. Afterward, we gauge the design stage reliability computation through the associated FTA, concluding that the product has significant improvement potential on both the hardware and software branches. Required modifications imply usage of a cloud computing platform, supplied by Amazon, as well as the approach of developing the platform with an LMS (Learning Management System). We complete this stage by delivering the architectural environment analysis in order to validate the aforementioned improvements and safeguard the functionalities outlined in the original project goals.

Chapter 4 deals with the implementation and verification stages of the software lifecycle. We highlight the fault taxonomy and tailored prevention methods, as well as the efforts undertaken to initialize the development of the e-learning application, alongside the progression of the unitary interface which underlies all advancement modules. We proceed with course development in accordance with the profile of a software quality assurance engineer and set up their hierarchy based on the user's technical level.

We presented the general principles upon which all testing must be based and proposed an exploratory testing and experience-grounded approach to efficiently identify and resolve potential issues. For the functional testing of our software system, we established in-depth conformance with previously defined specifications through a suite of test cases. We also verified platform interoperability with the existing LMS mobile application through scenarios which specifically target mobile-only functionality.

The topic of non-functional validation of computer systems is presented in **Chapter 5**. We introduce ways of establishing the security level of a software product through a risk analysis of potential vulnerability sources and their associated impact. We also expound upon types of attacks and attackers worth accounting for in such an analysis. We exemplified security-specific validation for our e-learning application through both manual testing procedures and specialized software-assisted scenarios, using Vega and ZAP. These can perform intrusion scan and vulnerability exploitation attempts based on

a list of known issues found in Web platforms. The analysis found that the platform's security level is high because the architectural environment is stable and the most severe issue we uncovered was minor.

We also proposed ways of optimizing the efficiency of usability testing through a selection of relevant devices matching their proportion in the worldwide mobile data usage report, as well as their operating resolution. Additionally, we presented auxiliary tools which facilitate usability testing, and which offer results relevant in the real world. Through their use, we identified and subsequently fixed several faults.

6.2. Original contributions

The original contributions are listed in a synthesized manner and ordered according to their presence within each chapter, as follows:

1. We performed a quasi-exhaustive analysis of the offerings within the Romanian public education system that pertain to software quality assurance, at the bachelor and Master's levels, and identified weaknesses with the theoretical content and geographical coverage required to produce experts in the field of computer software quality assurance.

2. We performed a SWOT analysis encompassing both the academic and online offerings provided by the likes of ISTQB with respect to obtaining the necessary knowledge. The conclusion was that practical study is not covered by either party and the way the content is structured is often poor.

3. Based on the SWOT analysis, we formed the hypothesis of creating an e-learning platform which resolves these issues by providing a flexible source of information, with accurate, easily digestible content. To this end, we compiled a survey in order to assess its feasibility and confirm the necessity of such a solution. This survey is comprised of 9 questions relevant to our goal: seven mandatory and two optional. We evaluated the results and portrayed the demands of each level of competence, with the authenticity of these results supported by values achieved with the survey's control question [E2].

4. Due to the fact that our hypothesis of creating an adjacent solution to solve existing issues has been proven viable, we continued with a QFD analysis facilitated by the House of Quality, which utilized our survey results to establish user requests and their impact coefficients [E2]. We could then construct the technical specification matrix on the premise of developing an application with a solid and flexible technical foundation, with a concise, consistent manner of presenting its content.

5. We performed a comparative analysis of existing methodologies, specifying their applicability based on the type of project and underlining the advantages and disadvantages which can derive from employing them [E1][B2].

6. Following this scrutiny, we put forth a method of optimizing the methodology and development processes for Web applications [B5]. This was built upon theoretical knowledge of software quality assurance, as well as experience working as a professional in the field to date. The method encompasses means of segmenting development, the mode of operation and structure of the team, the manner of planning and development time allocation, as well as processes required for day-to-day activities.

7. We recommended a blueprint for a sustainable team capable of running development projects for complex Web products [C5]. It derives its strengths from its multidisciplinary nature, an equal split of duties and decision-making ability, clear role separation, as well as great communication and knowledge sharing.

8. We analyzed the transition state diagram in accordance with the QFD results. This focuses on three access levels: administrator, instructor and student, but also on the available actions for a particular role. For student consumption of modules, we created a flow diagram based on their acquired level of knowledge. We also produced a Gantt chart highlighting the stages of content development for the learning modules.

9. We studied both the standards and best practice guides pertaining to Web security and identified the necessary adjustments required for achieving a secure and legally compliant product in accordance with EU regulations [A1][A3].

10. We compared testing techniques based on the level of applied verification and suggested use of a technique resulted from the interpolation of commonly used methods [E4]

11. We devised a best practice guide for developing a test plan which provides the most coverage for the product under test by prioritizing testing based on feature usage and facilitating a reduction in product time to market [E4].

12. We proposed solutions for improving both the security [A1][A3][C2], and the usability [B1][C1] of the product as early as the design stage, in order to minimize defect propagation and reduce the costs associated with issue remediation. These pertain to application and architectural environment security.

13. Starting with the premise of a computer system built from scratch, we performed a Fault Tree Analysis with the goal of determining the probability of an inability to function or limited access to the e-learning platform [B6]. Both software and hardware problems at the user and system level were considered. From the ensuing unfavorable results, we made two proposals for prototype improvement, culminating in much greater system reliability.

14. Based on the modifications employed to model its reliability, we performed a study of the architectural environment that accounted for both hardware solutions and the technical structure of the e-learning platform software.

15. We presented the fault taxonomy and devised preventative measures for each category. These are directly dependent on the development stage which introduces them and their quantifiable impact on the final product.

16. We applied the security recommendations identified pursuant to Web application development guides and EU legislation analysis from the project initiation stage [A1][A2][A3].

17. We constructed a unitary structure upon which the learning modules will be elaborated and displayed [E3]. This accounted for the improvement proposals for usability at this stage; we selected a color scheme which promotes focus and does not distract the user from the process of assimilation, and applied styles in a coherent, representative manner for each message meant for the user.

18. We compiled the profile of a quality assurance engineer based on the inherent qualities one must possess in this role. In order to determine if a user is able to successfully work in this field, we composed a test with 19 questions, detailed in Annex 4.

19. We elaborated the knowledge and study modules based on the requirements put forth by end users and synthesized through the QFD analysis. These were divided into two main trajectories, technical and quality assurance. Both offer notions which cover all knowledge levels and are offered in a manner assisted by practical examples and study verification questions.

20. We presented an experience-based approach to testing prior to the verification stage which correlates with the functionality list. This highlights a series of issues which can hinder the testing process and is able to reduce potential delays stemming from their required resolution.

21. We devised a series of test cases which ensure proper use of main functionalities from a student role perspective. These address the ability to access pages, to create a new user based on a preexisting configuration, as well as the ability to view material from the dedicated course catalogue.

22. Additionally, we structured the verification of main functionalities through two test suites. Conformity of the mechanism which allows module consumption in read-only mode for students and administering said modules in read-write mode for instructors, as well as the messaging functionality, was confirmed by the scenarios we designed.

23. Platform interoperability control with its mobile application was performed in order to ensure the solution's reliability and flexibility [B1][C1]. This included the analysis of mobile operating system versions used worldwide in order to select the most commonly used device types.

24. We studied the process of performing a risk analysis to determine the security vulnerabilities our solution can be exposed to, as well as potential areas of resolution [C6].

25. We determined the current security state of the platform through validation scenarios, both manual and automated, assisted by specific tools. In designing these test cases we employed studies of the top security attacks and guides on security-focused testing [C3][C4].

26. We proposed a solution for efficient Web platform responsiveness and compatibility testing, augmented by the study of worldwide mobile data traffic. This was enhanced by the selection of additional tooling which provides relevant results in establishing the level of usability of a Web application [E5].

6.3. List of papers

The research which led to the completion of this doctoral thesis can be found in the list of works (as author or co-author) published in IDB-indexed publications or proceedings of the international scientific conferences "Electronics, Computers and Artificial Intelligence - ECAI" (2017, 2018), "International Symposium on Advanced Topics in

Electrical Engineering - ATEE" (2017), "eLearning and Software for Education Conference - eLSE" (2017), "Quality and Dependability" (2016, 2018).

6.3.1. Articles in ISI indexed publications

[A1] S.D. Axinte, G. Petrică, I.C. Bacivarov, *GDPR impact on company management and processed data*, Quality - Access to Success, Vol. 19, No. 165, 2018, pp. 150-153, ISSN 1582-2559, WOS: 000450328700020.
[A2] G. Petrică, I.D. Barbu, **S.D. Axinte**, I.C. Bacivarov, I.C. Mihai, *E-learning platforms identity using digital certificates*, Proc. of the 13th International Scientific Conference "eLearning and Software for Education" Bucharest, 2017, Vol. 3, pp. 366-373, doi: 10.12753/2066-026X-17-228.
[A3] I.D. Barbu, G. Petrică, **S.D. Axinte**, I.C. Bacivarov, *Analyzing cyber threat actors of e-learning platforms by the use of a cloud based honeynet*, Proc. of the 13th International Scientific Conference "eLearning and Software for Education", Bucharest, 2017, Vol. 3, pp. 352-357, doi: 10.12753/2066-026X-17-226.

6.3.2. Articles in IEEE indexed publications

[B1] S.D. Axinte, I.C. Bacivarov, *Improving the quality of Web applications through targeted usability enhancements*, 10th International Conference on Electronics, Computers and Artificial Intelligence - ECAI 2018, 28 - 30 June 2018, Iași, România, doi: 10.1109/ECAI.2018.8679098, WOS: 000467734100167, Electronic ISBN: 978-1-5386-4901-5.
[B2] S.D. Axinte, G. Petrică, I.D. Barbu, *Managing a software development project complying with PRINCE2 standard*, 9th International Conference on Electronics, Computers and Artificial Intelligence - ECAI 2017, pp. 176-181, ISBN: 978-1-5090-6458-8, ISSN: 2378-7147, doi: 10.1109/ECAI.2017.8166435, WOS: 000425865900051.
[B3] G. Petrică, **S.D. Axinte**, I.C. Bacivarov, I.C. Mihai, M. Firoiu, *Studying cyber security threats to Web platforms using Attack Tree diagrams*, 9th International Conference on Electronics, Computers and Artificial Intelligence - ECAI 2017, pp. 154-159, ISBN: 978-1-5090-6458-8, ISSN: 2378-7147, doi: 10.1109/ECAI.2017.8166456, WOS: 000425865900072.
[B4] I.D. Barbu, C. Pascariu, I.C. Bacivarov, **S.D. Axinte**, M. Firoiu, *Intruder monitoring system for local networks using Python*, Proceedings of the 9th International Conference on Electronics, Computers and Artificial Intelligence - ECAI 2017, pp. 182-185, ISBN: 978-1-5090-6458-8, ISSN: 2378-7147, doi: 10.1109/ECAI.2017.8166457, WOS: 000425865900073.
[B5] S.D. Axinte, G. Petrică, I.D. Barbu, *E-learning platform development model*, The 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, 2017, pp. 687-692, doi: 10.1109/ATEE.2017.7905126, WOS: 000403399400134.

[B6] G. Petrică, I.D. Barbu, **S.D. Axinte**, C. Pascariu, *Reliability analysis of a Web server by FTA method,* The 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, 2017, pp. 683-686, doi: 10.1109/ATEE.2017.7905101, WOS: 000403399400133.

6.3.3. Articles in IDB indexed publications

[C1] S.D. Axinte, I.C. Bacivarov, *Enhancing the quality of a Web application through responsive design*, Proceedings of the 16th International Conference on Quality and Dependability, Sinaia, Romania, September 26th-28th, 2018, pp. 196-200, ISSN 1842-3566.
[C2] S.D. Axinte, G. Petrică, I.C. Bacivarov, *Study on the security of e-learning platforms*, Asigurarea Calitatii - Quality Assurance, Vol. XXIV, Issue 95, July-September 2018, pp. 26-28, ISSN 1224-5410.
[C3] G. Petrică, **S.D. Axinte**, I.C. Bacivarov, *SSL digital certificates analysis*, Asigurarea Calitatii - Quality Assurance, Vol. XXIV, Issue 93 (January-March 2018), pp. 22-27, ISSN 1224-5410.
[C4] S.D. Axinte, G. Petrică, I.C. Bacivarov, *Browsers cookies - an (in)security analysis*, IJISC - International Journal of Information Security and Cybercrime, vol. VI, no. 2 (December 2017), pp. 23-26, ISSN 2285-9225, doi: 10.19107/IJISC.2017.02.03.
[C5] S.D. Axinte, I.C. Bacivarov, *Maintenance testing of a software product*, Proceedings of the 15th International Conference on Quality and Dependability, Sinaia, Romania, 2016, September 14th-16th, 2016, pp. 237-243, ISSN 1842-3566.
[C6] S.D. Axinte, *Security challenges for software development companies*, IJISC - International Journal of Information Security and Cybercrime, vol. V, no. 2, 2016, pp. 9-16, ISSN 2285-9225.

6.3.4. Other published works

Book:
[**D1**] G. Petrică, **S.D. Axinte**, I.C. Bacivarov, *Dependabilitatea sistemelor informatice*, Matrix Rom, 2019, ISBN 978-606-25-0529-5.

Article:
[**D2**] G. Petrică, **S.D. Axinte**, *A comparative study on security of e-learning platforms in the Romanian academic field*, Considerations on challenges and future directions in cybersecurity, I.C. Mihai, C. Ciuchi, G. Petrică (editors), Sitech, 2019, pp. 19-26, ISBN 978-606-11-7004-3, eISBN 978-606-11-7005-0.

6.3.5. Scientific reports throughout the doctoral studies

[E1] Scientific report no. 1/2016, *Analiza modelelor de dezvoltare software.*

[E2] Scientific report no. 2/2016, *Colectarea și analiza cerințelor de business. Transformarea conceptelor în specificații tehnice pentru un produs software.*
[E3] Scientific report no. 3/2017, *Studiu analitic privind tehnologiile adecvate proiectării și implementării unei platforme Web.*
[E4] Scientific report no. 4/2017, *Structurarea activităților de verificare și validare. Elaborarea planului de testare ca etapă în managementul calității unui produs software.*
[E5] Scientific report no. 5/2018, *Testarea comportamentală a unei aplicații Web în conformitate cu cerințele non-funcționale.*

Selected bibliography

[1] W.A. Shewhart, Economic control of quality of manufactured product, Van Nostrand Company, 1931.

[8] R. Black, E. Van Veenendaal, D. Graham, Foundations of Software Testing: ISTQB Certification, 3rd Edition, 2012.

[30] First Instance of Actual Computer Bug Being Found, Computer History Museum, https://www.computerhistory.org/tdih/september/9.

[41] Ministerul Educației Naționale, Raport privind starea învățământului superior în România, 2016, https://www.edu.ro/sites/default/files/_fișiere/Minister/2017/transparenta/Stare_sup%20%202016.pdf.

[50] J.B. ReVelle, J.W. Moran, C.A. Cox, The QFD Handbook, Wiley, 1998.

[52] Agile Scrum Web Development, https://www.neonrain.com/agile-scrum-Web-development/.

[53] C. Gao, G.C. Hembroff, Implications of modified waterfall model to the roles and education of health IT professionals, DOI:10.1109/NOMS.2012.6212076, https://www.semanticscholar.org/paper/Implications-of-modified-waterfall-model-to-the-and-Gao-Hembroff/6ad11aed72ff31c0dbdaf8b9123b28b9bef422b7.

[55] S.D. Axinte, G. Petrică, I.D. Barbu, E-learning platform development model, The 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, 2017, pp. 687-692, doi: 10.1109/ATEE.2017.7905126, WOS: 000403399400134.

[57] S.D. Axinte, G. Petrică, I.D. Barbu, Managing a software development project complying with PRINCE2 standard, 9th International Conference on Electronics, Computers and Artificial Intelligence - ECAI 2017, pp. 176-181, ISBN: 978-1-5090-6458-8, ISSN: 2378-7147, doi: 10.1109/ECAI.2017.8166435, WOS: 000425865900051.

[59] S.D. Axinte, I.C. Bacivarov, Maintenance testing of a software product, Proceedings of the 15th International Conference on Quality and Dependability, Sinaia, Romania, 2016, September 14th-16th, 2016, pp. 237-243, ISSN 1842-3566.

[62] ISO/IEC 27002:2013 Information technology - Security techniques - Code of practice for information security controls, https://www.iso.org/standard/54533.html.

[64] ISO 27001:2013 - interpretarea în limba română.

[65] EUR-Lex - Access to European Union law, Directive (EU) 2016/680 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data by competent authorities for the purposes of the prevention, investigation, detection or prosecution of criminal offences or the execution of criminal penalties, and on the free movement of such data, and repealing Council Framework Decision 2008/977/JHA, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016L0680.

[66] S.D. Axinte, G. Petrică, I.C. Bacivarov, GDPR impact on company management and processed data, Quality - Access to Success, Vol. 19, No. 165, 2018, pp. 150-153, ISSN 1582-2559, WOS: 000450328700020.

[69] S.K. White, What is COBIT? A framework for alignment and governance, 2019, https://www.cio.com/article/3243684/what-is-cobit-a-framework-for-alignment-and-governance.html.
[70] ITIL® Foundation, ITIL 4 edition, AXELOS, 2019.
[73] A. Spillner, T. Linz, H. Schaefer, Software Testing Foundation - A Study Guide for the Certified Tester Exam, 4th Edition, 2014.
[75] IEEE 29119-5-2016 - ISO/IEC/IEEE International Standard - Software and systems engineering -- Software testing -- Part 5: Keyword-Driven Testing, https://standards.ieee.org/standard/29119-5-2016.html.
[76] IEEE 12207-2017 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes, https://standards.ieee.org/standard/12207-2017.html.
[77] V. Cătuneanu, A. Bacivarov, Structuri electronice de înaltă fiabilitate. Toleranța la defectări, Editura Militară, Bucureşti, 1989.
[78] C. Jones, Applied Software Measurement: Global Analysis of Productivity and Quality, 3rd edition, McGraw-Hill Education, 2008, ISBN 978-0071502443.
[82] D. Peters, Product Managers, do you know how much your bugs cost?, https://deanondelivery.com/product-managers-do-you-know-how-much-your-bugs-cost-72b6e36e7684.
[89] S.D. Axinte, G. Petrică, I.C. Bacivarov, Study on the security of e-learning platforms, Asigurarea Calității - Quality Assurance, Vol. XXIV, Issue 95, July-September 2018, pp. 26-28, ISSN 1224-5410.
[92] S.D. Axinte, I.C. Bacivarov, Enhancing the quality of a Web application through responsive design, Proceedings of the 16th International Conference on Quality and Dependability, Sinaia, Romania, September 26th-28th, 2018, pp. 196-200, ISSN 1842-3566.
[93] S.D. Axinte, I.C. Bacivarov, Improving the quality of Web applications through targeted usability enhancements, 10th International Conference on Electronics, Computers and Artificial Intelligence - ECAI 2018, 28 - 30 June 2018, Iași, România, doi: 10.1109/ECAI.2018.8679098, WOS: 000467734100167, Electronic ISBN: 978-1-5386-4901-5.
[94] G. Petrică, I.D. Barbu, S.D. Axinte, C. Pascariu, Reliability analysis of a Web server by FTA method, The 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, 2017, pp. 683-686, doi: 10.1109/ATEE.2017.7905101, WOS: 000403399400133.
[95] V.M. Cătuneanu, I.C. Bacivarov, Fiabilitatea sistemelor de telecomunicații, Ed. Militară, București, 1985.
[98] M. Bach, Most Reliable PC Hardware of 2016, https://www.pugetsystems.com/labs/articles/Most-Reliable-PC-Hardware-of-2016-872.
[99] A. Klein, Hard Drive Stats for Q2 2016, https://www.backblaze.com/blog/hard-drive-failure-rates-q2-2016/.
[101] Amazon Web Services (AWS) - Cloud Computing Services, https://aws.amazon.com.

[107] TalentLMS: Cloud LMS Software - #1 Online Learning Platform, https://www.talentlms.com.
[112] Amazon Data Centers, https://aws.amazon.com/compliance/data-center/controls/.
[113] GDPR Compliant eLearning Software - TalentLMS, https://www.talentlms.com/gdpr/.
[122] R. Patton, Software Testing, Sams Publishing, 2005, ISBN 978-0672327988.
[130] Positive Technologies, Web application vulnerabilities: statistics for 2018, https://www.ptsecurity.com/ww-en/analytics/web-application-vulnerabilities-statistics-2019/.
[134] ISO/IEC 27005:2011.
[135] OWASP Testing Guide v4, https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents.
[138] OWASP ZAP, https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project.
[143] Mobile & Tablet Browser Market Share Worldwide, Nov 2018 - Oct 2019, https://gs.statcounter.com/browser-market-share/mobile-tablet/worldwide/#monthly-201811-201910-bar.
[147] Uptime.com, Website Speed Test Result for qualityassurance.talentlms.com, https://uptime.com/freetools/website-speed-test/177990/qualityassurance.talentlms.com.