

THE DEVELOPMENT METHODOLOGY OF SOFTWARE PRODUCT BASED ON WEB ENGINEERING

METODOLOGIJA RAZVOJA SOFTVERSKOG PROIZVODA ZASNOVANOG NA WEB INŽENJERSTVU

Vedrana Nerić¹, Nermin Sarajlić²

Abstract: The need for the development methodology of software product is a consequence of the increasing use of software product based on Web engineering, as well as the need for such product is in accordance with scientific requirements and the rules of software engineering. This paper deals with the development methodology of software product based on Web engineering, development approaches, certain specifics in development, functional requirements, as well as the life cycle of modern Web application. The paper shows the use of Web engineering, development environment and development tools for Web applications. It gives an overview of the project team, who are members of the project team, what skills, knowledge and competencies are necessary for a successful Web project. The methodology is applied to a concrete software product.

Keywords: methodology, software product, Web engineering, Web development life cycle, Web application

Sažetak: Potreba za metodologijom razvoja softverskog proizvoda posljedica je sve većeg korištenja softverskog proizvoda zasnovanog na web - inženjerstvu, kao i potrebe da takav proizvod bude u skladu s naučnim zahtjevima i pravilima softverskog inženjerstva. Rad obrađuje metodologiju razvoja softverskog proizvoda zasnovanog na web - inženjerstvu, pristupe razvoju, određene specifičnosti prilikom razvoja, funkcionalne zahtjeve kao i životni ciklus moderne web - aplikacije. U radu je prikazana primjena web - inženjerstva, razvojna okolina i razvojna pomagala za web - aplikacije. Dat je prikaz projektnog tima, koji članovi čine projektni tim, koje su potrebne vještine, znanja i kompetencije za uspješan web - projekat. Metodologija je primijenjena na konkretnom softverskom proizvodu.

Ključne riječi: , softverski proizvod, web - inženjerstvo, životni ciklus web - razvoja, web - aplikacija

INTRODUCTION

In a relative short period of time, the Internet and the World Wide Web have become present in all fields of human activity, overcoming other technological developments in history. They also grew rapidly in the domain and scope of use, significantly affecting all aspects of our lives. Industrial branches, such as manufacturing, tourism, catering, banking, education, state administration, are now using the Web to improve their activities. Even traditional information systems and database systems have moved to the Web. A new wave of mobile Web applications appeared as a result of improvements in the area of wireless technologies and Web devices. Web-based systems are expected to be reliable and functional. In order to develop such systems, people who are dealing with this need a secure technology, a disciplined and repeatable process, better development tools, and a set of good working guidelines. The Web engineering area, which has recently started to

appear, meets these requirements. Web engineering uses the principles of science, engineering, software engineering and management, and approaches systematically to successful development, deployment and maintenance of Web systems and high-quality applications. Its goal is to put under control disorder in Web-based systems, minimize risks, and improve the quality and maintenance of Web locations. [1]

1. DEVELOPMENT OF WEB APPLICATIONS

1.1. Web

The World Wide Web, known as the Web, has become a very attractive and dominant platform for implementation of business and social applications and organizational information systems. It has also become a universal user interface for business applications, information systems, databases, and legacy systems. The Web supports the management of documents and workflows, cooperative work, distributed knowledge, and media sharing. Web-based systems and applications offer a variety of content and functionality to a large number of users and serve for many different purposes. As dependence and reliance

¹codeNest Ltd., Tuzla, BIH

²Faculty of Electrical Engineering, University of Tuzla, BIH
vedrana_neric@yahoo.com

Paper submitted: March 2017 Paper accepted: May 2017

on Web-based applications have increased significantly, so performance, reliability, quality, maintenance, and scalability of Web applications are becoming of great importance, and users expect Web applications to be more usable, more reliable, more secure, personalized. However, most Web developers do not recognize and do not take into account many ambiguous and unique Web application requirements and do not recognize that the features and requirements of the Web systems differ significantly from traditional software, including their development. Many Web development projects are implemented in ad hoc way and without the adoption of design methodology, resulting in a poorly designed Web system. To successfully create complex Web-based systems and applications, both large and small, Web developers need to adopt a disciplined development process and design methodology and use better development tools. Web engineering advocates a disciplined approach for successful Web development, taking into account the unique features and requirements of Web-based systems. The essence of Web engineering is successfully managing diverse and complex Web application development and avoiding potential failures that can have serious consequences. [2], [3]

1.2. Unique aspects of Web applications

Web applications have certain unique internal features that make Web development different and more challenging than traditional software development. The key features of Web applications are [2]:

- Most Web applications are evolutionary in their nature, requiring frequent changes of content, functionality, structure, navigation, presentation, or implementation. They are especially developed in terms of their requirements and functionality, especially after the system is put into use. For most Web applications, the frequency and degree of changes are much higher than traditional software applications, and for many applications it is not possible to list all the requirements at the beginning. Therefore, successful management of evolution, changes, and newer Web application requirements is the main technical, organizational and management challenge, that is more demanding than traditional software development.
- Web applications are intended to be used by a large, diverse, remote group of users, who have different requirements, expectations, and skills. Therefore, user interface and usability should meet the needs of various anonymous users. Furthermore, the number of users who access the Web application at any time is unpredictable and can vary significantly, creating performance problems.
- Web-based applications require the presentation of different content, such as text, graphics, images, audio and/or video content, where content can be in-

tegrated with procedural processing. Therefore, their development includes the creation and management of content and presentation in an attractive way, as well as providing subsequent content management continuously after the initial development and implementation.

- Web-based systems require a good aesthetic appearance, as well as easy navigation.
- Web applications, especially those intended for a global auditorium, need to support many different social, cultural, and national standards, including multiple languages and different unit systems.
- The security and privacy requirements of Web-based systems are generally more demanding than traditional software. Therefore, there is a growing demand for Web application security.
- Web applications need to deal with different display devices, formats and support of hardware, software and networks with different access speed.
- The consequences of the failure or dissatisfaction of users of Web-based applications can be much worse than conventional IT systems.
- Time frames for the development of Web applications are shorter, and that significantly affects the methodology and process of design and development.
- The spread of new Web technologies, standards and competitive pressure to use, brings its advantages and additional challenges for the development and maintenance of Web applications.
- Development nature of Web applications requires an incremental development process.

2. WEB ENGINEERING

Web engineering is the application of scientific, engineering and management principles and disciplined and systematic approaches to the successful development, implementation and maintenance of high-quality Web-based systems and applications [2]. It deals with the development and organization of new knowledge about Web application development and application of this knowledge to the Web application development and with solving new requirements and challenges faced by developers. Web engineering can also be viewed as a holistic and proactive approach to the development of complex Web-based systems. With a holistic approach, all aspects of the development process, functional requirements, operating environment of the application, IT support and other infrastructure, connections and interactions among them are identified, analyzed, implemented, and evaluated. Therefore, Web engineering deals with all aspects of the development of Web-based systems, starting from conception and development to implementation, performance evaluation and continuous maintenance. Web engineering covers a number of areas: requirements de-

tection and analysis, Web system modelling, Web architecture, Web design, Web site design, coding, interface with databases, and other Web-based systems, Web quality, Web usability, Web security, Web system performance evaluation, Web testing, Web development methodologies, Web development process, Web metrics, and Web project management. Contrary to the perception of some experts, Web engineering is not a clone of software engineering, although both include programming and software development. While Web engineering uses software engineering principles, it also includes new approaches, methodologies, tools, techniques, and guidelines to address the unique requirements of Web-based systems. In many aspects, the development of Web-based systems is much more than the development of traditional software. There are subtle differences in the nature and the life cycle of Web-based systems and traditional software systems, and in the way they are developed and maintained. The nature and features of Web-based applications and their development require Web engineering to be multidisciplinary, including inputs from different areas, such as system analysis and design, software engineering, hypermedia/hypertext engineering, requirements engineering, human-computer interaction, user interface, information engineering, indexing and information retrieval, testing, modelling and simulation, project management, graphic design and presentation. [2], [4]

2.1. Web development

The development of high-quality Web applications is not a coincidence. This requires a systematic plan for development and implementation, as well as architecture design, testing, evaluation, installation of security protection measures, and the adoption of implementation procedures. Design and analysis of Web systems, however, represent a significant challenge: systems need to be understood at many different levels of abstraction and viewed from many different perspectives. Modelling of Web systems and architecture play an important role in the development of today's complex Web systems. [2]

2.1.1. Web architecture

Web architecture can be defined as the fundamental organization of a system, unified in its components, their interactions, and relations with the environment, and the principles governing design and evolution. It uses abstractions and models to simplify complex structures. Architecture is a framework, describes the structure and makes the system understandable. It helps in the transition analysis to implementation. The Web subsystem architecture consists of [2]:

- the entire system architecture that describes the interaction of the network and different servers, such as Web servers, application servers, database servers

- an application architecture that describes different available information modules and functions
- a software architecture that identifies different software and base modules needed to implement the application architecture.

2.1.2. Web design

Design plays an important role in the development of high-quality Web applications with high performance. Web design should set many different requirements, some of which may be contradictory: design for usability – interface design, navigation, design for understanding, and design for performance – responses, design for security and integrity, design for evolution, growth and maintenance, design for testing, design for universal availability, design for localization, graphic and multimedia design, Web site design. [2]

2.1.3. Web security

As there are more and more applications from email, shopping, sales to banking and business interactions and their related data that are launched on the Web, hackers increasingly target Web applications. So, Web security is at the centre of attention. Information is one of the most important resources in almost every organization. After the internal networks of these organizations are linked to the Internet and the Web, they have become a potential target for cyber attacks. Data and information must be protected from unauthorized access, as well as from malicious corruption. A wide range of legal and regulatory issues surround Web development, including the need to protect sensitive business and user information. A good set of security procedures and practices will limit the violation of Web security and unwanted exposure of information. There are six security criteria for the development of Web applications based on empirical evidence [2]:

- active organizational support for security in the Web development process
- appropriate controls in the development environment
- visibility of security in all areas of the development process
- delivery of a cohesive system that integrates business requirements, software, and security
- fast, rigorous testing, and evaluation
- trust and responsibility.

2.1.4. Web testing and evaluation

Testing plays a key role in the overall Web development process. However, testing and evaluation are often neglected aspects of Web development. Many developers test the system only after the system meets the failures, or after the limitations that become evident, resorting to what can be called retroactive testing, and what is desir-

able in the first place is proactive testing at different stages of the Web development life cycle. Benefits of proactive testing include ensuring proper functioning, guaranteed performance levels, avoiding expensive retroactive corrections, optimal performance, and lower risk. Testing and validation of large complex Web systems is a difficult and expensive task. Testing should not be done just before the end of the development process. Developers and their managers need to review wider and follow a more holistic approach to testing – from design to implementation, maintenance and continuous improvement. Test planning should be performed at an early stage of the project life cycle. Test plan provides a road map so that the Web site can be evaluated through requirements or design reports. It also helps to estimate the time and effort required for testing, setting up the test environment, getting staff for testing, writing test procedures before testing can begin. Web testing can be grouped into the following categories: browser compatibility, display Web site, session management, usability, content analysis, availability, backup and recovery, transactions, order processing, operational business procedures, system integration, performance, login, and security. [2]

2.2. Web project team

Creating large Web systems requires knowledge from a wide range of areas such as Web programming, Web architecture, Web design, multimedia, Web security, Web performance, and usability. Since a small number of Web developers have all the necessary knowledge, the development of Web-based systems is no longer limited to an individual and must rely on a group of people who can work together on the development and implementation of Web-based system or Web application. So knowledge cooperation has become an important aspect of Web development. Web application development requires a team of people with different skills: programmers, graphic designers, Web site designers, usability experts, content developers, database designers and administrators, data transfer and networking experts, Web server administrators. The Web development team is multidisciplinary and must be more versatile than the traditional software development team. Due to the rapid development, to which the Web application project is particularly exposed, and due to the fact that the development of modern software is managed by a group of people, not more individuals, communication between the team members and their motivation and coordination by the project manager are among the most important success factors for the project. Teams formed for the development of Web applications are characterized by three important features [5]:

- Multidisciplinary: As the Web application consists of content, hypertext structure and presentation for ideally, a very wide auditorium, Web developers need to have knowledge from different fields.

- Parallelism: While the tasks of traditional software projects are divided into specific development aspects, Web projects are usually divided by problems. As a result, Web project team subgroups are similarly compiled with regard to their expertise, which means that parallel developments have to be coordinated. The communication effort is greater in the Web project teams than in the development of traditional software systems.
- Small size: Due to the short development cycles and often quite limited budgets, Web project teams are composed of a small number of team members (about six on average, rarely more than ten). Larger tasks are divided and done in parallel in sub teams.

3. LIFE CYCLE OF SOFTWARE PRODUCT BASED ON WEB ENGINEERING

With the advent of the Internet, the way software applications are developing through the Software Development Life Cycle (SDLC) has changed. Web-based business applications have gained popularity and wide acceptance due to their usability in terms of better functionality and mobility, enabling the connection of information around the world. Most companies develop business applications that are Web enabled to provide wide availability, both inside and outside the organization. The way Web-based business applications are developing should take into account the dynamic nature and characteristics of the Web environment. The new development methodology called Web Development Life Cycle (WDLC) includes all SDLC phases appropriately and is used for the development of Web-based business applications in a fast and structured way. This methodology is a hybrid of two methodologies – software development life cycle (SDLC) and prototype methodology. WDLC uses the components of each methodology combining them in a new approach that reduces development time, adds the structure to an unstructured problem, and keeps users involved in the entire development life cycle. The Web development life cycle can be defined as a systematic methodology that includes a more modern Web application development process according to specifications and Web standards as efficiently as possible. The same concept of the SDLC model is followed using the waterfall model. However, by introducing a prototype methodology, WDLC becomes an iterative process that involves users in the development process. Like traditional software development, the Web development process can also be divided into different phases of the life cycle. This can help to shape the team efficiently and standards and procedures can be adopted to achieve maximum quality. The methodology for the development of Web-based business applications includes the following phases: requirements phase, design phase, implementation phase, testing phase, and maintenance phase. [6], [7]

3.1. Requirements phase

The requirements phase is an early phase before the development of Web-based business application. This phase involves the gathering basic information about the application that needs to be developed. Information for graphic and functional application development is collected. Information is provided by users or clients and described in the requirements document. It discusses the type of information, its format, how information will be edited, the navigation through this information, identifies the purpose of the Web application and the required functionalities. This information is recorded and analyzed to generate stable requirements in order to avoid potential conflicts and discussions in the near future. At the end of this phase, it will be ensured that the proposed Web-based business application is feasible with the capabilities of the organization and development team and presented in a complete specification for other phases. When more information is obtained between the development team and users, the probability of obtaining a successful, Web-based business application is greater. [6]

3.2. Design phase

Design activities can be considered as a very important phase in the life cycle of development in which the skeleton of Web-based business applications is provided. At this phase, various diagram techniques are used to analyze and design functionalities of the application. In this way, the different Web application components will be broken down into smaller fragments. Then each of these components will be designed and developed later as if they were individual programs. After the functionality of the Web application and various components are designed, the developer will begin to create a physical prototype. Layout and navigation are usually designed as a prototype. All aesthetic properties are adjusted in the interface design. At this phase, database design, architecture design, logical and physical design are generated and documented in the requirements specification design. All necessary changes in the specification, the iterative process will hold for redesign activities. [6]

3.3. Implementation phase

This phase includes the creation of application in detail according to the approved design. Activities include creating a database, making the interface, and code implementation. Developers start creating prototypes for each component of the Web application. At this phase, the functionality of the Web application is being developed. Web developers work with users to identify the components of Web applications that are necessary for implementation. After identifying the critical components, they begin to develop a prototype for these components.

Users should be actively involved with developers when creating each component. Many times users know what they want, but they do not understand what can be done in Web development. Therefore, a Web developer should guide users and include them in the development process. When functionality is added to the Web application, the user needs to test the components and give feedback to the developer. Feedback and approvals can be delivered via online return forms, e-mail, fax, or phone. After all necessary component changes are made, the developer starts working on the next component and the same procedure is repeated. This creates the iterative process of this WDLC phase and increases the speed at which the Web application develops by working in modules. The prototype is usually developed on a test server or development server. This allows users to work with developers until the prototype is completed. Once the developed component is complete, the database and other files are moved to a production server. [6]

3.4. Testing phase

Unlike traditional software, Web-based applications require intensive testing, as applications will always function as a multi-user system with bandwidth limitations. Some of the tests that need to be done are integration testing, stress testing, scalability testing, load testing, security testing, browser compatibility testing, etc. Usability testing should also be carried out to ensure that the Web-based business application is error-free and meets the usability requirements. The completed Web-based business application should also ensure that there are no broken links and difficulties in using the application. [6]

3.5. Maintenance phase

Maintenance is continuous WDCL phase that never ends. Maintenance can include changes to existing programs, changes in appearance, functionality, or anything that needs to be done on the Web application after its implementation. [6]

4. USE OF THE METHODOLOGY ON THE EXAMPLE OF APPLICATION

As a practical example of use of the methodology, the software product – Web-based application named "Sales" has been developed. The application is intended for companies that sell different types of products. The application provides recording (create, read, update, and delete) of data on representative offices, sellers, salaries, buyers, products, and orders.

4.1. Requirements specification

Within the **requirements phase**, the first phase of the life cycle of the software product development – Web application "Sales", basic activities of the requirements engineering were carried out: finding and analyzing requirements, requirements documentation, validation – verification of requirements. An interview with the end user was conducted and the basic information needed for application development was collected. The purpose of the application, necessary functionalities, and different components for each part of the application were identified. An iterative and cooperative process of analysis, documenting the analysis results, and continuous verification of the correctness of requirements understanding were performed. The result of the requirements phase is described in the requirements specification for the application that is being developed. For each requirement from the specification, the type and priority of the requirement are specified. The main purpose of the requirements specification is to achieve a mutual agreement with regard to requirements on which software system will be developed further. The requirements specification is used as a basis for concluding a contract between the system client and the party that develops the system and as input to the next design phase. Usually, the requirements phase and the design phase overlap, so that during the setup of the requirements specification in the preliminary version begins with the design. A good part of the requirements is crystallised only at the beginning of the system design using these requirements. Only when it comes to thinking from the design perspective sets a series of new questions and requirements.

4.2. System modelling with UML diagrams

Within the **design phase**, the structure of the system, the components of which the system will consist, the way the components work, as well as the interface between components are formed. Therefore, a solution is designed that determines how the software product will work. Design is an iterative process, i.e. the design came with gradual refinement and elaboration through several iterations. For design the software product – Web applications "Sales" the following UML diagrams have been used: use case diagram, class diagram, and activity diagrams.

4.2.1. Use case diagram

After creating the requirements specification, functional requirements have been elaborated and modelled as use cases. Therefore, a use case diagram has been created (Figure 1), actors of the system have been identified (Table I) and a detailed specification in the form of a table for each use case has been made. An example of the specification of one use case Product Records is shown in Table II.

Table I: Actors

Actor	Semantic
Product manager	The system user who is responsible for recording (create, read, update, and delete) data about products, product groups, product types, sellers, representative offices, places, cantons as well as salary calculations.
Seller	The system user who is responsible for recording (create, read, update, and delete) data about buyers, places, cantons, and orders.
Buyer	A person who buys or orders products.

Table II: Specification of the use case UC6

Use case: Product records
ID: UC6
Actors: Product manager
Preconditions: 1. Collected data about product
Events flow: 1. Product manager enters the product code 2. If the same code exists The system reports a message that the code already exists and Product manager returns to Step 1. 3. Product manager enters basic information about product 4. If there is no data about the product group <product group records> 5. Product manager enters the product group code 6. If there is no data about the product type <product type records> 7. Product manager enters the product type code
Post-conditions: 1. Product information has been updated

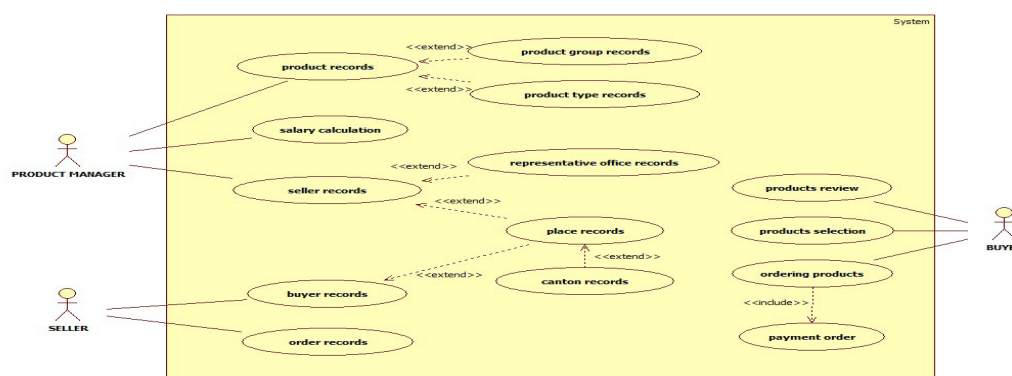


Figure 1: Use case diagram

4.2.2. Class diagram

When designing the system, the analysis was carried out – a fundamental and strategic activity that deals with the creation of a model that reflects the basic properties of the requested system in accordance with the requirements set. The results generated by the analysis are: classes at the analysis level, description of the classes that model key concepts within the given business domain, the realization of use cases, i.e. description of how system behaviour defined by use cases can be realized by interaction of objects from the described classes. The following results were obtained with further analysis: description of relationships between classes, grouping classes in packages, modelling more complex processes from the given business domain.

4.2.3. Activity diagrams

For the software product – Web application "Sales" activity diagrams for the main activities of the system are created. Activity diagrams describe the logic of procedures, business processes, and workflow. The flows of functionality presented in the use case diagrams are described by activity diagrams, which show all the activities that occur within the observed functionality. With one activity diagram, it is possible to display more potential scenarios that can occur when performing some functionality. They are used for modelling various types of processes and can join any model element to describe its dynamic behaviour.

4.3. ER diagram and database

The database refers to a company that sells different products (e.g. IT equipment, televisions, air conditioners, etc.). The database records data about representative offices, sellers, salaries, buyers, products, and orders. The ER diagram of database for the software product – Web application "Sales" is created, entities, relationships, and relational model are defined, based on which the Oracle relational database is created.

4.4. Application

Within the **implementation phase**, the designed solution is implemented with the help of the available programming languages and tools. Creation of database, creation of user interface and code implementation were done on the development server. The functionality of the application is developed by iterative prototyping. The end user was also involved in the development process, by testing the developed components and giving feedback to the developer. The software product – Web application "Sales" is created using Oracle development tools Forms Builder and Reports Builder and programming languages

PL/SQL and SQL. Toad for Oracle tool was used to create the database and the necessary functions and procedures. The "Sales" application is a Web application that is accessed via the Web using an intranet, private network of the company that will use the application. Unlike client – server applications that consists of the part that runs on the server and the part that need to be installed on the user's computer – client, Web applications do not need to be installed on the user's computer and instead of the executable file (.exe), a set of documents is generated which can be viewed using the Web browser.

By running the appropriate link in the Web browser, the login form for the application opens. After the Username and Password have been entered, pressing the Login button logs in to the application. Each user gets their username and password, which uniquely identifies them in the database. The login process ensures that unauthorized users can not use the application and data stored in the database. The user interface of the application consists of the initial form and menu that are displayed immediately upon application logging, the forms for recording (create, read, update, and delete) data, contained in the part of the menu Processing, and the parametric forms for running reports, and the reports, which the system produces and which are located in the part of the menu Reports.

When designing the user interface, the basic principles of design were applied. The most common system classification in three parts is used for easier handling of information and greater transparency: the upper part, intended for navigation, the middle part, the largest part connected to the user's work, and the bottom part showing information about what the user is currently doing (status line).

The part of the main menu Processing is divided into several sub-menus: Processing 1, Processing 2, Processing 3, Processing 4, and Exit the application. Within the sub-menu Processing 1 there are forms: Place, Canton, Representative office. Within the sub-menu Processing 2 there are forms: Product, Product group, Product type. Within the sub-menu Processing 3 there are forms: Buyer, Seller, and Salary. Within the sub-menu Processing 4 there is the form Order – Order items.

After running any form of the application, in the upper part of each form, there is the toolbar with the control buttons on which the interface icons are located. These buttons are for running interface actions. Some actions can also be run by pressing certain keys on the keyboard, as described in the meaning of the toolbar buttons. Although the toolbar is easy to use and users can easily assume the meaning of individual buttons, there are text descriptions that appear when passing the cursor over the icon in order to avoid misinterpretation and confusions. Within the user interface, the grammatical order "object – action" is

Seller, Salary, and all application reports, while Buyer and Order forms will be disabled in the menu.

Within the **testing phase**, it is verified whether the software product really works as intended. During and after the implementation, the software product – Web application “Sales” has been tested, eliminating the detected application errors. Syntax and logical errors are removed by debugging during application development. Testing is divided into several independent testing units. Independent testing units are software components that are obtained during the software system implementation phase. After unit testing, integration testing was performed to examine the interaction between separate tested units after their integration. The unit and integration testing of all basic components of the application confirmed their correctness and proper functioning. Functional testing is performed using a black box method. The performance of the specified functions and the accuracy of the output data based on the software requirements specification were analyzed, which verified that all specified requirements were provided by the application. The database is tested, as well as all the components of the program code. Value integrity was checked in database fields. The software design was tested with a white box method, which examined and analyzed the source code, checked all functions and procedures, and checked different combinations of program commands. In this way, it is verified that there are no infinite loops or code that is never executed. The functionality of all forms has been checked using CRUD (Create, Read, Update, Delete) tests to confirm the correctness of creating, reading, updating, and deleting all objects. An audit of the ability to run all reports and display the correct data on reports for different combinations of input parameters was performed.

The application is designed modularly with the possibility of efficient upgrading, so that new functionality can easily be added if needed by incorporating new forms and reports. Acceptability testing was conducted based on the end user specification and testing of usability by a future user who was involved in the process of the software product development. Browser testing was performed to determine that the application can run without any problems with some of the known browsers (Internet Explorer, Opera, Google Chrome). In order to determine that the system functions are available to the intended users, security testing has been performed. An application login for different types of users (product manager, seller) has been tested to determine that a user can access those parts of the application to which they are entitled and assure that those parts of the application, to which the user is not entitled are disabled.

As a result of this phase of the development life cycle, the testing phase, it has been determined that the software

product – Web application “Sales” meets the quality requirements (functionality, usability, ease of navigation, security, maintainability), meets the specification and needs of users.

After completion of the development of all necessary components of the application, completion of testing by developers and users, and all necessary changes on the components, the database and other files are placed on the production server where the database layer (Oracle Database 11g) and the application layer (Oracle Application Server Forms and Reports Services) were installed for the software product – Web application “Sales”.

After the introduction of the software product – Web application “Sales” in use, it is necessary to carry out end user training. Users on training be acquainted with the way that the application runs and works, and they receive explanations for all the terms and concepts to work with forms that they will use and reports they will run. Within the training, to the user is described in detail the user interface, as well as the way of recording or performing the operations such as create, read, update, delete, and search data. Illustrative documentation, obtained by describing all use cases by means of forms and reports, offers a user manual that will answer any question about the system and thus facilitate the operation of the user.

After the introduction of the software product – Web application “Sales” in use, the software will continue to be repaired, modified, and upgraded within the **maintenance phase** in order to meet the needs of users. Software changes are necessary to keep pace with new user requirements, business environment changes, hardware advancements, etc.

5. CONCLUSION

A brief history and lack of structure in the Web application development have led to many unsuccessful examples. Given the fact that software products based on Web engineering are gaining in importance and are increasingly being used, and in order to avoid unsuccessful examples, it is necessary to have disciplined approach to the development, and structural methodology, which will help developers to successfully implement Web projects. In this paper the development methodology of software product (Web application) based on Web engineering, which is in accordance with the standards, the scientific requirements and the rules of software engineering is processed. The methodology is applied to a particular software product – Web application “Sales”. In this paper, through a systematic approach, adding structure to the development process is described and applied to the concrete

example. Such an approach to the development process can provide a model for organizations involved in development. The model can be used to create a Web presence, whether it is information Web site, e-commerce, or intranet systems with the possibility of reporting. Structured methodology can: facilitate developers to develop complex Web-based systems, minimize development risks and serve as a help to ensure successful development. Web engineering should continue to be developed in order to deal effectively with the unique challenges posed by Web-based systems, so further research in this field should be focused on finding new methods, techniques and other approaches to the challenges of developing Web-based systems.

REFERENCES

- [1] A. Ginige, S. Murugesan: Web Engineering: An Introduction, IEEE Multimedia, 2001
- [2] G. Rossi, O. Pastor, D. Schwabe, L. Olsina: Web Engineering – Modelling and Implementing Web Applications, Springer – Verlag, London, 2008
- [3] E. Mendes, N. Mosley: Web Engineering, Springer – Verlag Berlin Heidelberg, 2006
- [4] W. Suh: Web Engineering – Principles and Techniques, Idea Group Inc, 2005
- [5] G. Kappel, B. Proll, S. Reich, W. Retschitzegger: Web Engineering – The Discipline of Systematic Development of Web Applications, John Wiley and Sons, Heidelberg, Germany, 2006
- [6] A. Abdul-Aziz, A. Koronios, J. Gao, M. Suhaizan Sulong: Towards Effective Development of Web-based Business Applications, IBIMA Publishing, 2012
- [7] D. Howcroft, J. Carroll: A Proposed Methodology for Web Development, IS Research Centre, University of Salford, 2000

BIOGRAPHY

Vedrana Nerić graduated in 2006 from the Faculty of Electrical Engineering of the University of Tuzla, department of Technical Informatics. During the studies, as a recognition of the achieved success, she was awarded with three Silver and a Gold Medal by the University of Tuzla. She acquired the academic professional title of a Master of Electrical Engineering at the same Faculty in 2013 and she is currently a PhD student. She is employed at the codeNest Ltd. Tuzla, as a software engineer. In 2014, she was selected to the collaborative teaching assistant for the scientific field of Computer Science at the Faculty of Electrical Engineering of the University of Tuzla.

Nermin Sarajlić was born in Tuzla, Bosnia and Herzegovina in 1965. He graduated from the Faculty of Electrical Engineering, University of Tuzla, Bosnia and Herzegovina in 1987. He acquired a Master of Sciences degree at the Faculty of Electrical Engineering and Mechanical Engineering, University of Tuzla, Bosnia and Herzegovina in 1997, and acquired a Ph.D. degree at the Faculty of Electrical Engineering, University of Tuzla, Bosnia and Herzegovina in 2002. His field of interest is the calculation of the coupled electromagnetic-temperature fields; Cryptography, cryptanalysis.