JOURNAL OF SOFTWARE ENGINEERING & INTELLIGENT SYSTEMS
ISSN 2518-8739
30th April 2019, Volume 4, Issue 1, JSEIS, CADMEI Copyright © 2016-2019
www.jseis.org

THEMATIC KNOWLEDGE FRAMEWORK ON HUMAN FACTOR IN SOFTWARE MAINTENANCE PRACTICE: A STUDY IN A MICRO SOFTWARE COMPANY

[1] ZELJKO STOJANOV

[1]University of Novi Sad, Technical Faculty "Mihajlo Pupin", Zrenjanin, Serbia
Email: zeljko.stojanov@uns.ac.rs

ABSTRACT

Due to the recognized constraints faced by micro software companies, knowledge management of human related issues becomes important for their efficiency, quality of their products and services, and competitiveness at the market. These companies do not have resources to implement the best practice proposals and international standards in their practice, and therefore, the most suitable way to assess their practice and organize knowledge is by using inductive participatory approaches that start from the real context. This article presents a lightweight inductive method for identifying and systematizing knowledge, and its implementation in a micro software company aimed at developing a thematic knowledge framework on human factor in software maintenance practice. The development of the thematic framework is based on joint work of researchers from the university and the programmers from the company, with the full support and commitment of the company management. The thematic framework contains themes related to programmers and software users as the main actors in software maintenance practice. The benefits for the company, the implications for practitioners from industry and researchers, and further research directions are also discussed.

Academic Editor: Manoranjitham Muniandy, Universiti Tunku Abdul Rahman, Malaysia

Published: 30th April, 2019

Funding: The research is funded by Ministry of Education, Science and Technological Development, Republic of Serbia under project number TR32044.

Article Type: Research Article

Keywords: human factor; software maintenance; knowledge identification; knowledge systematization; thematic framework; micro software company;

1. INTRODUCTION

Human work and knowledge have been recognized as complex and problematic factors influencing the practice in software engineering [1], with significant contribution to individual performance, organizational capabilities, innovativeness and competitiveness of software organizations [2]. Grenčík and Legát [3] classified people skills as the key performance indicators in maintenance. Production and maintenance of reliable software require software engineers with appropriate set of knowledge, skills and experiences, which are not related only to technical issues, but also to project and product management, design, quality, customer expectations, legal issues, intellectual property and strategies [4]. Software experts are usually labelled as knowledge workers, whose work is characterized with creative thinking, creativity, openness to changes and challenges [5]. In addition, Olaisen and Revang [6] suggested that knowledge management activities should be planned, managed and controlled in software teams, and based on empirical study they pointed out the importance of changing roles in teams for sharing knowledge.

The fact that humans are more complicated and less predictable than software cannot be ignored, which places software engineering among the most difficult disciplines performed by humans [7]. Although it has been recognized that people dominantly influence software maintenance activities, human factor have not been comprehensively investigated in literature. For example, Lenberg et al. [8] reported that less than five percent of software engineering publications are concerned with the 'soft' or human-related topics. Even more, most software practitioners consider human factor as marginal for the practice [7].

Most of the recent studies on human factor in software engineering have used qualitative or mixed methods, which enables deeper understanding of human and social issues. In this direction, Lenberg et al. [8] developed a thematic framework containing challenging issues for directing further research in software engineering from industrial perspective, with the focus on the individuals, teams and groups. The main challenge in improving practice is to identify and extract knowledge that resides in software organizations and make it available to all

Corresponding Author: Zeljko Stojanov
Email Address: zeljko.stojanov@uns.ac.rs

employees [9][10]. Sultan et al. [11] proposed the rules and the framework for tacit knowledge extraction which support transfer of tacit knowledge from experts to all other employees, which is essential for functioning of teams in software organizations and leads towards continual learning and improvement [12]. Chergui et al. [13] proposed an ontological model for acquiring tacit knowledge, which considers actions or personal attitudes in a given organizational context and uses explicitation and selfconfrontation techniques for transforming tacit knowledge in usable form for sharing and reuse. According to Rus and Lindvall [14] there is a need to identify and systematize knowledge in software organizations because it resides in people heads, and it can be lost if people leave software organization. Hwang et al. [15] stressed the importance of using formal and reliable sources of knowledge in organizations, which assumes identification and transformation of knowledge in a form suitable for use and sharing. In addition, Khosravi et al. [16] suggested that software companies should consider management of customer knowledge in order to improve software quality, reliability, as well as development and maintenance activities.

Software maintenance has been perceived as a set of complex and demanding activities requiring different kinds of knowledge [17]. According to IEEE Std. 1219-1998 [18], software maintenance is a set of activities aimed at modifying software product after delivery in order to correct faults, improve performance or to adapt to a modified environment. April and Abran [19] suggested that human factor, organizational and management issues are the most critical for efficient software maintenance. Solving maintenance tasks requires maintainers with knowledge about the application domain, the specifics of an organization using the software, and variety of software engineering techniques and tools. Anquetil et al. [20] argued that the fundamental problem in software maintenance is the lack of various types of knowledge since it is usually only in heads of software engineers. Therefore, identification, systematization and reuse of knowledge can help maintainers to more efficiently cope with difficulties in everyday maintenance activities. Since software maintainers rarely search for new knowledge, but rather rely on previously acquired knowledge [21], the reuse of the knowledge and available information is one way to improve the efficiency of maintenance activities and to decrease maintenance costs [22][23]. This assumes that knowledge is structured, generalized and stored in a form that enables easy access and use. However, in many cases the knowledge is lost through iterative informal refinements of skills and methods previously implemented in the practice [24].

This article presents a thematic knowledge framework on human factor in software maintenance practice, which is developed in a micro software company by using a lightweight method for identifying and systematizing knowledge. The method is implemented in a micro software company. The method is based on the real organizational needs and does not follow any prescribed model for assessing the processes and organizing knowledge. The knowledge was systematized as a thematic framework containing themes and sub-themes related to characteristics of the programmers and clients. The article is structured as follows. The next section presents literature review on human factor frameworks in software maintenance and knowledge management practice in small software companies with the focus on managing knowledge related to humans. The third section outlines proposed inductive lightweight method for knowledge identification and systematization. The fourth section contains description of a case study in a micro software company, while the fifth section presents the research findings. The sixth section includes discussions on the research findings, benefits for the company, implications for the practitioners and researchers, limitations of the study and trustworthiness. The last section contains concluding remarks and future research directions.

2. RELATED WORK

This section outlines related work in the field of knowledge management practice on human factor in software maintenance, with the focus on knowledge management frameworks on human factor and knowledge management in small software companies.

Adoption of knowledge management principles with the focus on human factor can significantly improve software evolution especially in the maintenance phase of software life cycle [25]. Effective use of knowledge in software maintenance assumes identifying sources of knowledge and extraction of useful information that is necessary for performing maintenance tasks [26]. In addition to studies that have analysed specific aspects of human involvement in software maintenance, some studies presented more general frameworks containing segments dedicated to human factor. For example, in the framework developed by Haworth et al. [27], one of the four main components, named *programmers*, deals with the human issues in software maintenance. Kitchenham et al. [28] presented a general software maintenance ontology containing a *peopleware* ontology dealing with people issues. Kitchenham et al. distinguished two types of staff involved in software maintenance: maintenance organization staff, and customer and user staff. Dias et al. [22] addressed human factor with *skills* subontology in their ontology for organizing knowledge used in software maintenance. In their ontology, Ruiz et al. [23] defined *agents* subontology with two types of agents: (1) automatic agents or software tools, and (2) human agents. Human agents can be individuals (people) and organizations. Human agents were defined with the concept of roles which allows generalization and flexibility for different situations in maintenance practice. Anquetil et al. [29] proposed

an ontology of the knowledge needed for performing software maintenance, in which human factor was addressed in a sub-ontology called the *Computer Science Skills*. Ulziit et al. [30] investigated the factors specific to managing global software maintenance process. Several management challenges and mitigation strategies were identified and classified under four general categories structured as a conceptual framework. The study was based on a systematic literature review and five interviews with industry practitioners. Human issues were addressed in the category *people* that includes factors such professional skills and experience, as well as personal abilities and behavioural characteristics.

However, all these models and frameworks were developed in order to direct empirical research in software maintenance and are suitable for larger organizations that have resources for implementing all required steps for assessing and improving their practice. At the other hand, small software companies have limited resources (particularly financial and human resources) and require tailoring software engineering processes and practice to their needs [31], usually do not follow international standards and prescriptive models for assessing and improving their processes [32], do not use adequate tools, techniques and methodologies for managing knowledge [33], and encounter uncertainties and organizational constraints to follow the prescribed knowledge management guides and success paths [34]. Hutchinson and Quintas [35] argued that majority of SMEs manage knowledge in an informal and deliberate way, which means that SMEs simply manage what they know. Small software companies tend to use knowledge acquired from both internal and external sources [36], which assumes existence of knowledge in systematized form within the companies. Further, O'Connor and Basri [37] indicated that small software companies usually perform everyday practice in informal way without following strict rules and with clear focus on humans and communication issues. In order to overcome these difficulties Lin [38] suggests that small companies should be dedicated to full diffusion of knowledge management practice, implemented as a series of stages from the initial evaluation of knowledge management activities, to formal adoption and institutionalization in regular daily activities.

Several aspects of human factor related knowledge have been researched in the context of small software companies. For example, after identifying problems with learning from past experience in small Norwegian software companies, Dingsøyr et al. [39] proposed a lightweight method for collecting knowledge in small software companies based on post-mortem reviews and experience reports after completing projects. This method is highly dependent on human skills and knowledge. In five case studies implemented in small Canadian software firms the following processes that enable creation of knowledge in small hi-tech firms are identified [40]: formal meetings, informal communities, project teams, external interaction, information technology-tools, and rapid prototyping that enables knowledge creation through action. In a multistage empirical study with software experts from very small software companies O'Connor and Basri [37] identified the following major issues in knowledge management practice in very small software companies: communication, learning and sharing, documentation and knowledge management process, and commitment. All these issues revealed high importance of human factor in software companies.

Presented literature review revealed that there is a need to more thoroughly investigate human factor in software engineering industrial settings, especially in the field of software maintenance that consumes majority of costs in software life cycle [17][19] and requires software engineers with complex set of knowledge and skills since maintenance tasks are generally more complex than development tasks [41]. This statement is particularly applicable to small software companies that make up over 50% of the software industry [42] and suffer from time, budget and human restrictions for investigating and improving their practice [43]. This strongly justifies the need for inquiring human factor issues in software maintenance in small software companies and developing systematized knowledge that will be helpful for practitioners from industry and researchers.

3. A METHOD FOR KNOWLEDGE IDENTIFICATION AND SYSTEMATIZATION

This section briefly outlines the basic assumptions and characteristics of the ***Lightweight Inductive Method for Knowledge Identification and Systematization (LIM4KIS)***. The method was created to facilitate the identification and systematization of knowledge in micro organizations (teams). European Commission [44] defines a micro organization as an organization up to ten members. The main concerns in creating the method are: implementing knowledge identification and systematization activities without disturbing the everyday practice in an organization and encouraging the adoption of knowledge management culture in everyday practice. The method assists an organization to investigate the practice and systematize the actual knowledge within a thematic knowledge framework. Inductive thematic analysis proposed by Braun and Clarke [45] is used as a method for the data analysis and development of the thematic knowledge framework.

The identification of knowledge starts with the investigation of the real state of the practice in an organization, without attempting to fit the research process and findings to any prescribed strategy, directive, standard or guideline. Since the organization staff have the best insight into the everyday practice, the method uses their knowledge and experience for the identification and systematization of the most relevant knowledge. The method

requires the full commitment of the organization's management in all phases of the research, which ensures availability of necessary resources and increases validity of the findings. The main characteristics of the method are:

- *It is inductive*. It starts from the real organizational context and incrementally develops knowledge about the observed practice, ensuring that the knowledge framework reflects the real state of the practice.
- *It is participative*. It involves the organizational staff in all phases of the research, from the research design to the validation of the findings (created thematic knowledge framework).
- *Frequent feedback*. The main research instruments for analyzing data, discussing current state of the research and providing feedback to the relevant staff are feedback sessions. The sessions are organized as working meetings in an organization, allowing staff and researches to work jointly and direct further activities based on the current state of the research.
- *Triangulation of data sources*. The method uses various data sources and the appropriate data analyzing techniques, which increases the validity of the findings. The sources of the data are interviews with the staff, observations of everyday practice, the transcribed records from the feedback sessions, the documents available in the organization, and historical data available in electronic repository.

Figure 1 presents an overview of the main phases, activities and outputs of the LIM4KIS method. The design of a study based on the LIM4KIS assumes careful planning of all activities and people engagements.

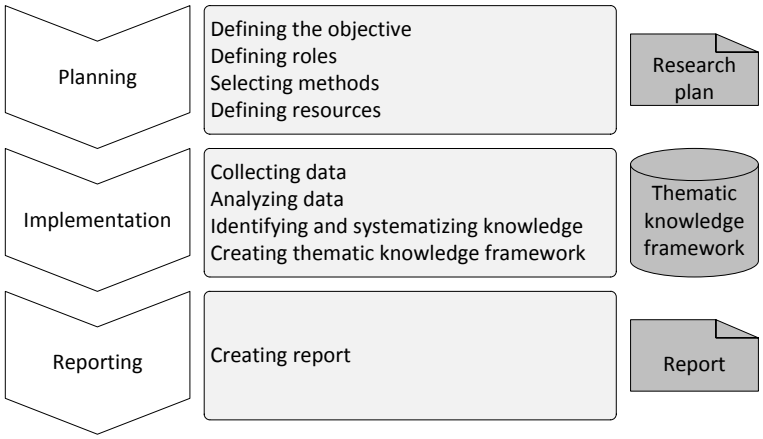

Figure. 1 Overview of LIM4KIS method

3.1 Planning phase

The planning of a study based on the LIM4KIS method includes determining the activities to be performed, and people involvement in these activities. The planning is based on the joint work of the management structures in the organization and the researchers. All plans are clearly presented in a document, which is available to the staff and the management of the organization, as well as to the researchers. The planning activities include:

- *Defining the research objective*. This is the first and the most important step during the planning. The objective is defined by the organization management and it should be aligned to the organization business strategy. The common case is to identify the research objective within the practice improvement projects conducted in the organization.
- *Determining the roles of the researchers and the organization staff*. Careful planning of the staff involvement ensures their availability and access to all necessary resources within the organization. In addition, this ensures that the most adequate people will be involved in all planned activities.
- *Determining the sources of the data*. In order to increase the validity of the study findings, variety of data sources are planned, which is based on their availability within an organization. The data sources may vary in different organizations. Generally, the most comprehensive understanding of the practice assumes combining a variety of different data sources and using both qualitative and quantitative methods in specific phases of the research [46]. Qualitative data originate from semi-structured interviews with the staff, practice observations and documents available in an organization, while quantitative data originate from documents and digital databases.
- *Selecting the methods used for collecting and analysing data*. The main data analysis method for the identification and systematization of knowledge is inductive thematic analysis [45], while some

quantitative data analysis methods are used for clarifying developed concepts during the feedback sessions [47].

Development of a thematic knowledge framework is primarily based on qualitative data sources, which are the most suitable choice for providing the most comprehensive and the most detailed evidence about the everyday practice [48][49].

3.2 Implementation phase

The identification of the themes by using LIM4KIS method entails gathering relevant data from the organization by using the interviews with the staff, the observations of the everyday practice, the documents, and by extracting the historical data about the practice from the electronic databases. In order to achieve the best results, the interviews are planned as exploratory and semi-structured, while the practice observations include writing field notes and tape recording of the usual conversations during the work.

The key activities in the method implementation are the feedback sessions, which are organized as working meetings in the organization. The participants for the sessions are invited based on the current state of the research (e.g. an interviewee whose transcribed interview is prepared for the discussion, or an observed employee). The sessions' discussions are tape recorded, transcribed and analysed by using inductive thematic analysis. Based on the analysis of the sessions' outputs, new data collecting and analysing cycles can be initiated. The iterative process of identifying and systematizing knowledge is presented in Figure 2.

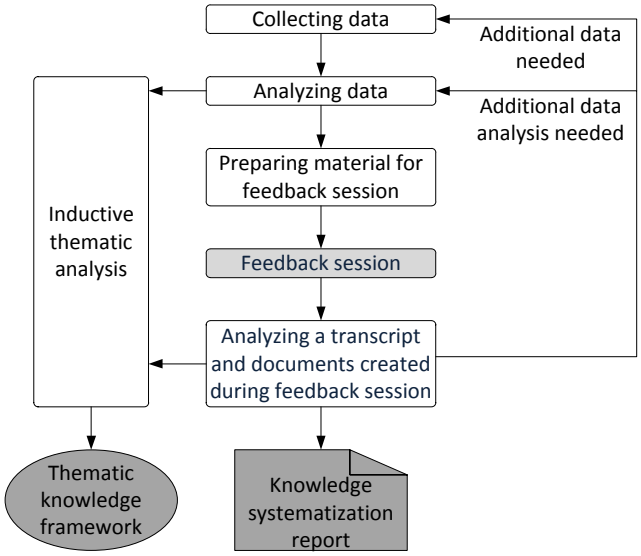

Figure. 2 Iterative process of creating a thematic knowledge framework by using LIM4KIS method

The development of a thematic knowledge framework assumes active involvement of the staff in the thematic data analysis. The main aim of involving the staff is the refinement of the identified concepts into the meaningful themes, sub-themes and sub-themes topics, and their integration into the thematic framework. Actually, only staff in the organization know what is going on in everyday practice and can describe each theme with the most appropriate words.

The final output of the method is a thematic knowledge framework, with the high-level themes relating to the essential issues in the practice. The sub-themes and sub-theme topics enable finer hierarchical organization of the knowledge, which is important for more precise description of investigated segment of the practice.

3.3 Reporting phase

The reporting phase includes writing the report on the whole research, which includes the detailed description of a developed thematic knowledge framework. The researchers write the report, while the organization management validate the documents. This is the simplest phase because all research findings are refined through feedback sessions during the second phase of the research with active involvement of staff in the organization.

4. CASE STUDY

The LIM4KIS method was implemented in a micro software company with seven employees. The staff include six programmers, and one technical secretary. The company manager has 23 year of industrial experience, while two leading programmers have over 15 years of industrial experience. Three other programmers have between one and four years of industrial experience. The young programmers face problems related to familiarizing with

the technical environment and adopting the company business policies. This clearly highlights the need for systematized knowledge in order to retain it in the company, and to make it available to all employees, especially to the novice ones.

The company develops and maintains over 30 business software applications for over 100 clients in Serbia. Trend analysis of clients' requests revealed that over 84 percent of the tasks are focused on maintenance activities [50]. This fact points out to the importance of software maintenance for the overall business operation of the company, which clearly motivated this research study aimed at systematizing the knowledge related to software maintenance. Due to the small number of the employees, which are completely dedicated to the regular daily tasks, the company management decided to conduct knowledge identification and systematization through joint work with the researchers from the university.

Knowledge identification and systematization activities were systematically implemented based on the proposed research plan. However, since the study was planned as qualitative, the design was flexible regarding the decisions how to organize activities in order to achieve the proposed objective. Quantitative data are used only to complement qualitative data by providing the evidence on trends in established maintenance activities. The plan only proposes the people roles, the needed resources, the methods for collecting and analysing data, and the estimations of the time frame for completing the research.

4.1 Roles

The main roles in the LIM4KIS method implementation are:

- *The leading researcher*, who is responsible for: designing the method and a study which implements the method, collecting data, preparing and leading the feedback sessions, analysing data, and systematizing the knowledge in the thematic framework. This role is assigned to the author of this article.
- *The project manager*, who is responsible for: setting the research objective, designing the method and a study which implements the method, ensuring the availability of all necessary resources, data analysis, and validation of the thematic knowledge framework and final research report. This role is assigned to the organization manager.

All other staff in the company, as well as other researchers are engaged according to the implementation plan that is defined and tailored to the company. The programmers were engaged as subjects for providing raw field data, and for data analysis where appropriate. The researchers were engaged for data analysis based on their expertise.

4.2 Data sources and data collecting activities

Although knowledge discovery and systematization activities were based on qualitative data analysis techniques (inductive thematic analysis [45]), data sources include both qualitative and quantitative data. All collected data, qualitative and quantitative, were prepared and discussed during the feedback sessions. The transcripts from these sessions also served as the sources for discovering and systematizing the knowledge about the maintenance practice. The following data sources were used: (1) interviews with the employees, (2) observations of the practice (3) company documents, and (4) records from internal repository of tasks and clients' requests. All data sources, data collecting and analysing activities are presented in Figure 3.

The first data collecting activity in LIM4KIS method implementation was the initial interview with the company manager. This interview served as a basis for planning and conducting the next data collecting activities: the interviews with other programmers, the observations of the practice, and extraction of the specific data from the local repository of tasks and clients' requests. The following data collecting activities were organized:

- Initial semi-structured in-depth interview with the company manager (about 60 minutes). The interview was used for guiding preparation of other data collecting activities.
- Semi-structured in-depth interviews with two leading programmers (about 45 minutes), and semi-structured in-depth interviews with other three programmers (about 30 minutes). The analysis of the interview transcripts was used for capturing the important aspects of the practice that deserve more attention during the observations.
- Observations of the programmers during the usual daily activities. Observations were conducted during the period of six weeks, 4 hour per day. The observations included taking field notes and recording usual conversations between programmers, as well as between programmers and clients. In addition, the programmers were asked to do screen shots of some maintenance activities in order to include them in the field notes.
- Additional semi-structured in-depth interviews with two leading programmers and the company manager lasted about 45 minutes, which are initiated after the practice observations.

- Collecting the company documents related to organizational issues, such as distribution of responsibilities, lists and short descriptions of software products, and the list with the clients.
- During the practice observations, the programmers were asked to extract some quantitative data from the internal repository of clients' requests and associated tasks. This type of data is in the computer science literature known as historical data (recorded data about the past activities) [51]. These data were extracted by using the purposeful SQL scripts, and imported in MS Excel for further quantitative analysis. These data were used for supporting the concepts identified by using thematic analysis.

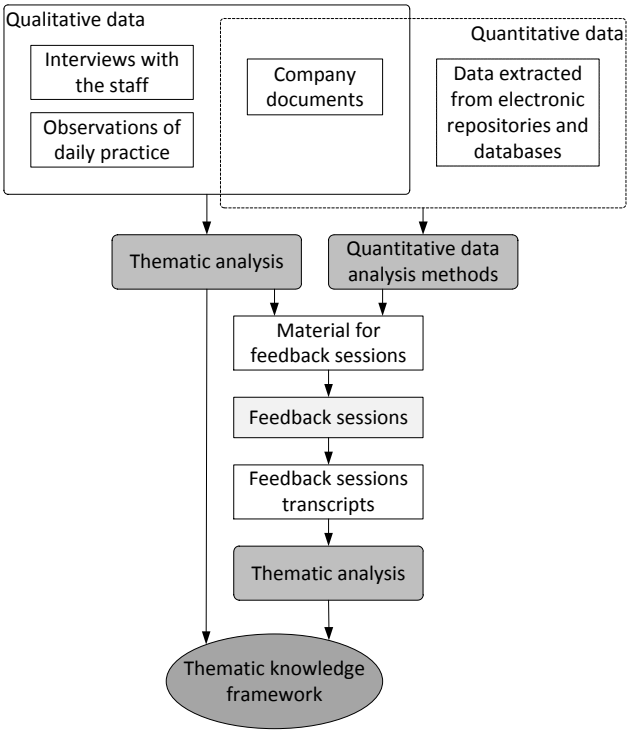

Figure. 3 Data sources and analysis methods in LIM4KIS method implementation

All data collecting activities were performed by the leading researcher, while data analysis involved other researchers based on their expertise, and proposed research plan. All collected data were prepared for the feedback sessions. The feedback sessions allow all interested parties (the staff and the researchers) to discuss current results and direct further research activities. The feedback sessions were organized between data collecting and analysing activities in order to assist in discussing and solving identified issues, as well as in directing further research activities.

4.3 Thematic data analysis

Identification and systematization of knowledge about human factor are based on the analytic methods for conducting inductive thematic analysis proposed by Braun and Clarke [45]. The researchers and the programmers from the company were involved in the data analysis. The researchers analysed collected data and provided feedback to the relevant programmers and the company management, according to the actual state of the research. The programmers had the opportunity to actively participate in the data analysis through discussions of the current research findings during the feedback sessions.

The data analysis was conducted by using several tools. All qualitative data were prepared as MS Word documents and imported in MAXQDA software tool for qualitative data analysis (http://www.maxqda.com/). Initial coding was completely done by using MAXQDA. Created code system was exported as a document in the rich text format supported by MS Word. The code system and memos were printed and prepared for the analysis in the feedback sessions. Most of the analyses related to the development of the themes was done on the papers by using pens in different colours. An example of a scanned paper with the analysis is shown in Figure 4. In this way, the programmers had the opportunity to actively participate in the development of the thematic framework. Through the discussions during the feedback meetings initial codes were developed into themes. The phases in data analysis were:

- *Familiarizing with the data*. The identification and systematization of knowledge was implemented as a sub-project within the SPI project [47], which enables the researchers to incrementally deepen

understanding of the practice and familiarize themselves with the organizational context in the company and with the data.

- *Generating initial codes*. The initial coding was completely done by the leading researcher in MAXQDA software. The initial codes were identified within unstructured text prepared from the field notes, interviews' transcripts and sessions' transcripts. All identified codes were discussed with the company manager and the leading programmers. In addition, for each initial code a short theoretical memo was written with the researchers notes about the code and the associated text segment, which is helpful for further analysis of initial code [52].
- *Searching for themes*. The focus of the analysis was shifted on searching for the broader themes that will enable organizing different codes into the potential themes. During the feedback sessions the researchers and the programmers used pens with different colours for highlighting the way of thinking, as it is presented in Figure 4.
- *Reviewing themes*. Themes were reviewed by the company manager and two leading programmers, who proposed further summing up of themes and introduction of sub-themes and sub-theme topics.
- *Defining and naming themes*. This phase started with a comprehensive review of the thematic framework in order to check the names and the content of all themes. Where appropriate, quotations from the empirical data illustrate the presented concepts, which is the common practice in qualitative research [53].

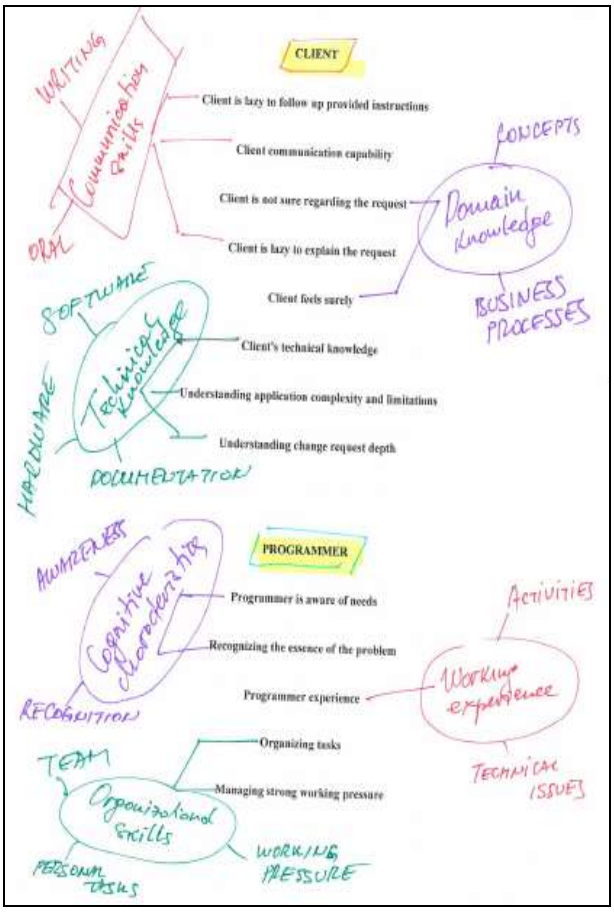

Figure. 4 Scanned document with the analysis of the themes during feedback sessions

During the whole research process reflexive memo writing helped in distilling and implementing all research activities and developing qualitative research findings [54]. Two types of memos were written: methodological memos – used for discussing methodological and organizational issues during the research, and theoretical memos – used in data analysis as the main tool for describing initial codes, themes and sub-themes, and the relationships between them in the developed framework. Theoretical memos are fundamentally important for developing theoretical framework as the main research finding of the study [52].

5. RESEARCH FINDINGS

Research findings were developed as a thematic framework, which is based on the evidence about the organizational context, individuals and their actions. According to the typology of the qualitative findings,

proposed by Sandelowski and Barroso [55], the findings of this study can be classified as Conceptual/Thematic description, since the findings are expressed as a set of developed themes integrated in the thematic framework. The author is aware that the thematic framework represents the current state of the practice, which could be modified and updated according to the needs that may arise in the future.

There are two types of people involved in maintenance processes: the staff in the software company, and the staff in the clients' organizations. These people communicate during all stages in a typical software maintenance process. The complexity of the maintenance tasks requires significant engagement of people. The effective communication requires that people possess the characteristics that will ensure reaching the objectives with the optimal efforts and costs. Effective communication between the programmers and the clients enhances knowledge-sharing success, which enables more efficient processing of maintenance requests [56]. The first group of people in this framework includes the programmers. The notation programmer is accepted in this framework due to the clear preferences towards this term by the company employees. The second group of people relates to the staff in the clients' organizations. Based on the years of experience, the company has developed the internal business policy requiring that only selected users from clients' organizations can submit maintenance requests. Therefore, client characteristics relate to these selected persons.

The final thematic framework for human factor in software maintenance practice is presented in Figure 5, which will be used for guiding the presentation of the research findings in the following subsections. Two main themes are *Programmer characteristics* and *Client characteristics*.

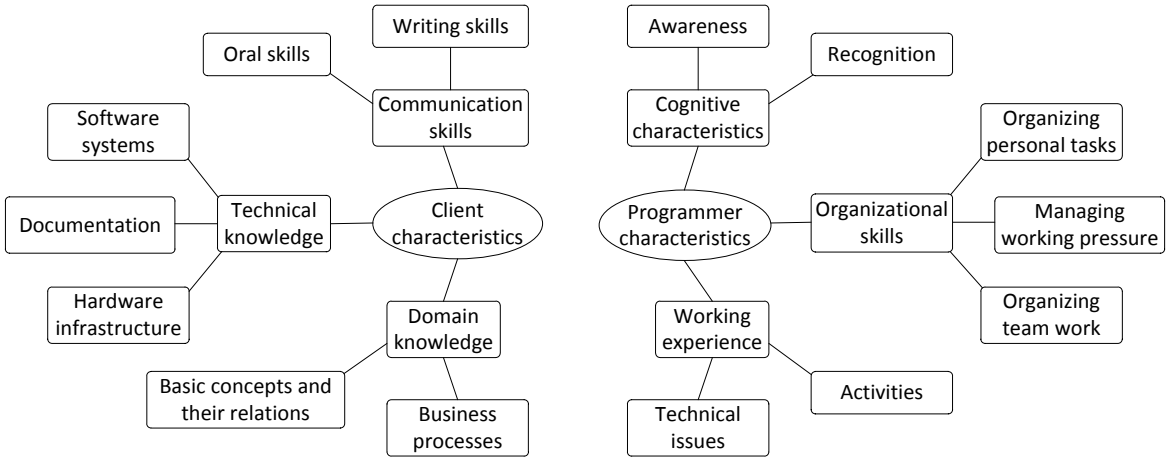

Figure. 5 Thematic framework presenting human factor in software maintenance

5.1 Programmers

Programmers play the key role in maintenance activities, acting as individuals or in teams. Their personal characteristics significantly contribute to the quality and timing of the maintenance activities. Thematic analysis revealed the following characteristics (sub-themes) of the programmers: cognitive characteristics, working experience, and organizational skills. These characteristics are interrelated. For example, organizational skills help programmers to organize their daily tasks related to maintenance requests and other activities within the company. However, this planning activity highly depends on programmers' working experience based on managing the similar situations in the past, and cognitive characteristics required for assessing the costs, risks, priorities and complexities of different tasks.

5.1.1 Cognitive characteristics

Cognitive characteristics relate to the ways of processing information and learning based on them during the everyday practice. Practically, this is an intrinsic characteristic of any programmer, which can be developed in the course of work. The following characteristics were identified (sub-theme topics): awareness and recognition.

Awareness relates to the ability to understand the real clients' needs and the real processes (being aware), which increases the effectiveness of programmers' own activities. This characteristic is very important because in many cases the clients are not aware of the real problems and the real costs of processing their requests. Awareness relates to artefacts (the complexity a software product or components affected by a maintenance request), people (characteristics of a client that submits a maintenance request, and characteristics of other programmers that may help in solving problems), and processes and tasks (knowing which methods and tools to use, or the arrangements of the steps).

Recognition relates to the ability to recognize the problems and surrounding issues, which includes perception, reasoning, and judgment in a given context. The main difficulties in maintenance relates to understanding the structure of software and the real cause of the problems, which requires perception and reasoning skills. A programmer should have perception about the software use in a client's business environment, and adequate reasoning about the steps and estimations in solving the problem.

5.1.2 Working experience

Working experience relates to the ability to react and process data in a given context, based on a subjective knowledge gained through programmer's involvement in everyday activities in the company. The programmers often use the word experience while describing their everyday activities, especially when talking about their productivity and efficiency. Of course, the experience was also mentioned in a negative context, especially when it is related to young programmers with the lack of experience in certain situations (e.g. when a programmer should extract more information about a maintenance request from a client). The experience has not been addressed in a systematic and organized way. This means that until this project, the experience has not been articulated and defined in any document or stored in a database system. The programmers relay on the experience stored in their minds. Regarding the practical issues, the following types of experience can be distinguished (sub-theme topics): technical issues and activities.

Technical issues experience includes experience in using: (1) software development and management tools (e.g. integrated development environments and internal software application for managing tasks and user requests), (2) programming languages (e.g. C#, scripting languages) and database management systems (e.g. SQL Server, MS Access), (3) libraries, APIs and various frameworks, (4) platforms (different operating systems, Microsoft .NET platform), and (5) technologies (e.g. Internet technologies, embedded systems). Due to the large number of daily tasks and constrained resources, the programmers usually have experiences with different technical artefacts, which is determined with the complexity of their tasks and years of experience. The main problem is the obsolescence of technical artefacts and technologies, which causes that some experiences have become obsolete and useless in the current maintenance practices.

Activities relates to the particular activities and tasks. This assumes recognizing that the current activity or task is similar to those experienced in the past and recalling how they were implemented. Typical cases are: (1) identifying the impact of change stated in a request (analytic steps of identifying software elements affected by change), (2) estimating costs and efforts for implementing all tasks identified in a requests (human resources, amount of work, possible risks, deadlines and milestones, costs), (3) modifying a software product (changing code, testing changed components, and verifying modifications), (4) deploying a changed software product in a client business environment (shutting-down required services, installing a software, configuring a software, validating software functionality), and (5) providing explanations to a user how to solve a problem alone (recalling all required steps from previous experience, guiding a user step-by-step, and explaining how to verify that the problem has been solved).

5.1.3 Organizational skills

Organizational skills enable the programmers to better organize themselves and achieve higher level of efficiency in their work. These skills are equally important when a programmer works alone or as a team member. The following organizational skills were identified (sub-theme topics): organizing personal tasks, managing working pressure, and organizing team work.

Organizing personal tasks relates to organizing daily tasks and planning tasks at the week and month levels. It assumes several practical skills such as: (1) accurate identification of tasks, with possible creation of a to-do list for each task, (2) ability to estimate costs, efforts and necessary resources for a task, (3) reliable prioritization of tasks by taking care about tasks importance and urgency, and (4) scheduling tasks in larger time periods (a week or a month).

Managing working pressure assumes recognizing the working pressure and finding out how to overcome it. Work under working pressure is quite common situation in the company because of the excessive workload caused by the large number of client requests. In these situation a programmer should make the right decisions regarding: (1) the appropriate scheduling and prioritization of tasks in his daily and weekly timetable, and (2) deciding which tasks can be redistributed to other programmers who can accept additional tasks.

Organizing team work is necessary for solving more demanding requests which include several tasks that cannot be solved by one programmer in a proposed deadline. The main issue relates to recognizing the situations requiring team work, and after that: (1) accurate identification of tasks and sub-tasks, (2) distribution of tasks to other programmers based on their availability, knowledge and skills, (3) coordinating and organizing work of

involved programmers, and (4) ensuring that all tasks are completed in proposed deadline and in accordance to specification.

5.2 Clients

Clients' characteristics were identified based on the programmers' opinions and experiences while working with them. The main concepts related to the client side of software maintenance practice in the company are: client organization - an organization that bought the software and uses the maintenance services, client or user - a person that uses software products and submits maintenance requests. The terms client and user are interchangeably used by programmers in the company, and based on the agreement with the company manager, in this thematic framework will be used the term the client.

The following characteristics of the clients were identified (sub-themes): communication skills, domain knowledge and technical knowledge. These characteristics are interrelated. For example, programmers' experience with the clients revealed that the lack of domain knowledge about the business processes in a clients' organization will negatively impact the communication and may cancel the benefits of possessing excellent communication skills. In other words, clients' communication skills, can be without any positive effect if they do not have adequate domain and technical knowledge about their business environment.

5.2.1 Communication skills

Communication skills are essential for obtaining efficient maintenance support. The communication between the programmers and the clients occurs repeatedly during the processing of requests. The most important moments during the maintenance request process requiring communication with the clients are: submission of a request, clarification of the request's details, approval of proposed tasks and costs, getting agreement about implementation details. The following communication skills were identified (sub-theme topics): writing skills and oral skills.

Writing skills relate to ability to write requests and provide other required information for processing requests in a written form.

Oral skills relate to ability to conduct a conversation in a direct contact, by using a phone or a Skype, which is sometimes necessary for discussing maintenance requests and request processing issues in more details.

In most cases, a typical maintenance process includes both written and oral communication. Due to the poor communication caused by the lack of communication skills or knowledge about a reported issue, some requests are postponed and or even discarded without notifying the clients. According to the programmers, the main characteristics of satisfactory communication skills of clients are:

- *Completeness* - a message should contain enough information for processing a maintenance request;
- *Conciseness* - a message should be short, but it should convey the essence of a request;
- *Clearness* - message should contain the appropriate words, and properly constructed sentences that convey the correct meaning;
- *Being focused* - careful reading or listening a message from a programmer improves understanding of what the programmer expects.

5.2.2 Domain knowledge

It is expected that all clients have enough knowledge about their business, i.e. that they possess domain specific knowledge. The domain knowledge is reflected in understanding the basic concepts and their relations in the domain, as well as understanding the business processes. The following problems occur due to the lack of the domain knowledge: (1) unclear specification of requests, (2) inability to provide the clarification of a request based on a programmer's inquiry, (3) delay in the processing of a maintenance request, or even rejecting a maintenance request, and (4) problematic installation and configuration of software in client business environment. The following areas of domain knowledge were identified (sub-theme topics): basic concepts and their relations, and business processes.

Basic concepts and their relations. Software applications are used for managing the entities and their relations in a business domain. These entities are named and regularly used by the clients in everyday practice. Due to the fact that domain knowledge has been already integrated into developed software application, it is reasonable to expect that maintenance activities will also depend on terminology used in business domain. The clients should know and understand all basic concepts and their relations in order to being able to specify meaningful maintenance requests. For example, if there is a relation between two concepts within a business domain, there will be the relation between the appropriate software items (e.g. between two classes in object-oriented design and two corresponding tables in database).

Business processes range from the level of tasks to the high-level processes in an organization. However, the clients usually do not have adequate understanding of business processes in their organizations, resulting with wrong descriptions of tasks, which leads to a confusing or meaningless maintenance request. A typical problem is when a task requires a new parameter, and a client does not know how to describe the new parameter, or even is not aware what modifications are necessary for introducing the new parameter in a software application that supports task execution. In addition, changes in one business process may require changes in other processes (e.g. changes in production will cause changes in finance), resulting with modifications of the appropriate software modules.

5.2.3 Technical knowledge

The complexity of software systems installed in clients' business environments is caused by the nature of the business processes they support. These systems include a variety of very complex software applications, and hardware components ranging from database and backup servers to sensor networks and controllers installed in complex industrial and agricultural environments. However, in most cases the clients have access to limited set of functionality and equipment, based on responsibilities and tasks assigned to them in their organizations. Only the clients working in the information technology sectors in larger companies have enough technical knowledge for recognizing the complexity of identified problems and maintenance requests. The following areas of technical knowledge were identified (sub-theme topics): software systems, hardware and communication infrastructure, and documentation.

Software systems. In most cases, the clients have poor technical knowledge about the software applications, which leads to the problems in their use, and in providing relevant information for the implementation of the appropriate maintenance actions. For example, the programmers are aware that a change in a graphical user interface may cause several and more complex changes in business and database layers, while the clients are not aware of this aspect of requested change. For most clients, this is a simple „cosmetic" work on the user interface. Of course, this leads to underestimating the costs and timescales for the implementation of appropriate maintenance tasks.

Hardware and communication infrastructure. Software applications require hardware and communication equipment for execution. In most cases software applications are distributed and have different users distributed in different departments in client organizations. In some cases, the clients submit requests that are not the real maintenance requests, which mostly happen when the problem is caused by the hardware or communication equipment. Therefore, the basic knowledge about hardware and communication infrastructure is necessary for understanding the potential sources of problems when using software systems.

Documentation. The introduction of any software application in a client business environment includes organization of trainings, as well as delivery of appropriate documentation (manuals and guidelines). However, the users are lazy to read the documentation, and they call programmers for any single problem. In most cases, they do not use software properly because they have not read the documentation. This results with totally meaningless requests, and actually reflects the users' laziness and the lack of understanding of their own work. In most cases, the programmers easily recognize these requests and provide additional instructions to users how to read the documentation or how to cope with the problems.

6. DISCUSSIONS

Human factor thematic framework reflects the human and social aspects of the maintenance practice in the company. This area becomes more and more important since it has the major impact on the productivity and the quality of maintenance activities and overall business performance of the company. Developed thematic framework is grounded in the data collected in the company, and therefore represents the real knowledge about the practice. Presented themes were discussed in detail, but without directly addressing the relations between them. However, the identified characteristics influence each other, which deserve additional discussion. Some typical situations related to the developers' characteristics will be shortly discussed, while the discussion on the clients' characteristics require their deeper involvement in the research and will be addressed in future work.

Working experience significantly influence the development of cognitive characteristics and organizational skills. For example, by experiencing a new situation in the practice, a programmer will store some information in his head, and will be able to recognize the similar problems and situations when encounter them later. This new problem or situation, and the solution increase the recognition ability of a programmer, but also help a programmer to better organize himself when encounter similar problems later.

The development of cognitive characteristics influences acquiring and improving necessary organizational skills. For example, by improving awareness, a programmer will be able to better predict possible problematic situations with client maintenance requests, which will lead to better organization of his own work or work of

other programmers if he needs their help. In addition, this cognitive development, and improvement of organizational skills will be added to the programmer's personal experience.

The next important issue relates to the programmers' ability to learn from their clients by solving their maintenance requests. Through the communication with the clients, the programmers learn about their business domain. For example, by acquiring some knowledge about the clients' business domain, a programmer will be able to recognize the hidden details of a maintenance request, which are not perceived by the client. This situation improves the recognition ability of the programmer and enables him to decompose a request into smaller parts that can be solved independently. In this way a programmer develops its cognitive characteristics, as well as organizational skills in solving such kind of situations. This obviously becomes a personal experience of a programmer, ready to be used in future.

6.1 Benefits for the company

The following benefits for the company can be perceived from this research:

- *Creation of the knowledge base about human factors*. The knowledge base was created in the form of a thematic knowledge framework containing the knowledge specific for the human side of the maintenance practice in the company. This knowledge base is important for young programmers which do not have enough experience in people related issues, especially in communication with the clients which is necessary in software maintenance. In addition, with this knowledge base, all relevant knowledge was captured and saved for the reuse in the company. The knowledge base can be easily updated according to the new needs.
- *Increased satisfaction and the sense of the personal importance in the company*. Programmers actively participated in all phases of the research and significantly contributed to the final thematic framework. They provided valuable assistance during the construction and validation of the thematic framework, which motivated them to work more efficiently and increased their satisfaction and the sense of the personal importance in the company.
- *Adoption of knowledge management practice and culture in the company*. This project is the first organized endeavour related to knowledge management activities in the company. Active involvement of the programmers ensures the adoption of knowledge management as a basis for further practice improvements in the company.

6.2 Implications for the practitioners from industry and researchers

This empirical evidence can be used as a source of knowledge for the practitioners from the industry. Although this thematic framework contains the knowledge specific for the selected software company, human issues tend to be similar in other small software companies. In addition, the practitioners may find the lessons how to organize a study in their organizations aimed at identifying and systematizing their specific knowledge, as well as the instructions on how to collaborate with the researchers in order to ensure the use of appropriate research methods and tools.

The researchers can draw several lessons from this research: (1) how to organize a study by using inductive qualitative research methods for discovering the knowledge in a specific organizational context, (2) how to cooperate with the staff within an organization in order to ensure the real findings about the everyday practice, and (3) how to organize the knowledge identification and systematization activities within a larger projects in the selected organizations (e.g. within process assessment and improvement projects). Furthermore, the presented human factor thematic framework can be used as a starting point for more detailed inquiring of human and social issues in other organizations.

6.3 Trustworthiness

The validity and rigor of qualitative research is based on ensuring that trustworthiness criteria, such as credibility and transferability are respected [57][58]. The credibility, or internal validity, of the study was ensured through careful application of inductive thematic analysis techniques and rich description of the context, research process and findings. Triangulation of various data sources [59] and active participation of all employees in the research process and validation of the created knowledge framework increased credibility of the study [60].

However, the researcher is aware about some threats to the trustworthiness. The first threat is dependability, which relates to inevitable changes in the context (the company), that will alter some segments of the investigated practice. All these changes should be reflected in the thematic knowledge framework. This threat will be addressed in further work directed towards ensuring the evolution of the knowledge framework that complies with the changes in the company.

The main threat to the trustworthiness of this study is transferability of the research findings. However, the aim of this study is not to provide the findings relevant for all similar software organizations, but rather to provide guidelines how to organize a study that will result with the thematic knowledge framework on human factor in specific organizational context. LIM4KIS method is presented in detail allowing the use in any small organization independently of its business areas.

7. CONCLUSION

This article presents a novel method for identifying and systematizing knowledge and a study aimed at identifying and systematizing knowledge on human factor in maintenance practice in a micro software company. The knowledge about human factor in software maintenance was identified and systematized by using inductive thematic analysis with active involvement of the company employees.

The main contributions of the presented research are: (1) a method for knowledge identification and systematization is presented, (2) an approach to increasing the validity of the research by involving the study participants in all phases of data analysis is presented, and (3) the developed framework can be used by practitioners and researchers as a starting point for investigating specific issues related to human factor in software maintenance. In addition, the presented method can be easily tailored for investigating role of human factor in maintenance activities in other types of small engineering organizations.

Further work includes several directions. The first one relates to development of a software tool for managing thematic knowledge frameworks, with the focus on knowledge evolution caused by dynamic nature of human factor in maintenance. In addition, investigation of human factors in software maintenance practice from clients' point of view can provide valuable insights that can be used for improving practice and developed thematic framework. And finally, the implementation of this method in other small knowledge-based companies is also promising research direction that will justify its usability.

ACKNOWLEDGEMENT

Ministry of Education, Science and Technological Development, Republic of Serbia, supports this research under the project "The development of software tools for business process analysis and improvement", project number TR32044.

REFERENCES

1. Schneider, K., Experience and Knowledge Management in Software Engineering. 2009: Springer-Verlag Berlin Heidelberg. DOI: 10.1007/978-3-540-95880-2.
2. Mao, H., S. Liu, J. Zhang, and Z. Deng, Information technology resource, knowledge management capability, and competitive advantage: The moderating role of resource commitment. International Journal of Information Management, 2016. 36(6, Part A): p. 1062-1074. doi: 10.1016/j.ijinfomgt.2016.07.001.
3. Grenčík, J., and V. Legát, Maintenance audit and benchmarking - search for evaluation criteria on global scale. Eksploatacja i Niezawodnosc - Maintenance and Reliability, 2007. 3(35): p. 34-39.
4. Feldt, R., L. Angelis, R. Torkar, and M. Samuelsson, Links between the personalities, views and attitudes of software engineers. Information and Software Technology, 2010. 52(6): p. 611-624. doi: 10.1016/j.infsof.2010.01.001.
5. Lotko, M., and A. Lotko, Cluster analysis of knowledge workers assessment of occupational threats and attitudes to character of work. Eksploatacja i Niezawodnosc - Maintenance and Reliability, 2015. 17(1): p. 80-89.
6. Olaisen, J., and O. Revang, Exploring the performance of tacit knowledge: How to make ordinary people deliver extraordinary results in teams. International Journal of Information Management, 2018. 43: p. 295-304. doi: 10.1016/j.ijinfomgt.2018.08.016.
7. Capretz, L.F., Bringing the Human Factor to Software Engineering. IEEE Software, 2014. 31(2): p. 104. doi: 10.1109/MS.2014.30.
8. Lenberg, P., R. Feldt, and L.G. Wallgren, Human factors related challenges in software engineering: an industrial perspective. In Proceedings of the 8th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE'15). 2015. p. 43-49. doi: 10.1109/CHASE.2015.13.
9. Alagarsamy, K., S. Justus, and K. Iyakutti, The Knowledge Based Software Process Improvement Program: A Rational Analysis. In Proceedings of International Conference on Software Engineering Advances (ICSEA 2007). 2007. p. 61-67. doi: 10.1109/ICSEA.2007.73.
10. Galvis-Lista, E., and J. M. Sánchez-Torres, A Critical Review of Knowledge Management in Software Process Reference Models. Journal of Information Systems and Technology Management, 2013. 10(2): p. 323-338. doi: 10.4301/S1807-17752013000200008.

11. Sultan, Z., R. Abbas, S.N. Bhatti, and A.M. Jadi, Role of visualization in tacit knowledge: extraction and applications in software requirements. Journal of Software Engineering & Intelligent Systems, 2017. 2(1): p. 62-69.
12. Lindvall, M., I. Rus, and S.S. Sinha, Software systems support for knowledge management. Journal of Knowledge Management, 2003. 7(5): p. 137-150. doi: 10.1108/13673270310505449.
13. Chergui, W., S. Zidat, and F. Marir, An approach to the acquisition of tacit knowledge based on an ontological model. Journal of King Saud University - Computer and Information Sciences, 2018. In Press. doi: 10.1016/j.jksuci.2018.09.012.
14. Rus, I., and M. Lindvall, Knowledge management in software engineering. IEEE Software, 2002. 19(3): p. 26-38. doi: 10.1109/MS.2002.1003450.
15. Hwang, Y., H. Lin, and D. Shin, Knowledge system commitment and knowledge sharing intention: The role of personal information management motivation. International Journal of Information Management, 2018. 39: p. 220-227. doi: 10.1016/j.ijinfomgt.2017.12.009.
16. Khosravi, A., A.R.C. Hussin, and M. Nilashi, Toward software quality enhancement by Customer Knowledge Management in software companies. Telematics and Informatics, 2018. 35(1): p. 18-37, doi: 10.1016/j.tele.2017.09.007.
17. Bourque, P., and R.E. Fairley (ed.), Guide to the Software Engineering Body of Knowledge (SWEBOK), 3rd ed. 2014: IEEE Press. Piscataway, NJ, USA.
18. IEEE Std 1219-1998. IEEE Standard for Software Maintenance. 1998: IEEE. New York, USA.
19. April, A., and A. Abran, Software Maintenance Management Evaluation and Continuous Improvement. 2008: IEEE Computer Society & Wiley. Hoboken, NJ, USA. 2008.
20. Anquetil, N., K.M. de Oliveira, K.D. de Sousa, and M.G.B. Dias, Software maintenance seen as a knowledge management issue. Information and Software Technology, 2007. 49(5): p. 515–529. doi: 10.1016/j.infsof.2006.07.007.
21. Ramal, M.F.N., R. de Moura Meneses, and N. Anquetil, A disturbing result on the knowledge used during software maintenance. In Proceedings of the 9th Working Conference on Reverse Engineering. 2002. p. 277-286. doi: 10.1109/WCRE.2002.1173085.
22. Dias, M.G.B., N. Anquetil, and K.M. de Oliveira, Organizing the Knowledge Used in Software Maintenance. Journal of Universal Computer Science, 2003. 9(7): p. 641-658. doi: 10.3217/jucs-009-07-0641.
23. Ruiz, F., A. Vizcaino, M. Piattini, and F. Garcia, An Ontology For The Management Of Software Maintenance Projects. International Journal of Software Engineering and Knowledge Engineering, 2004. 14(3): p. 323-349. doi: 10.1142/S0218194004001646.
24. Deridder, D., A Concept-Oriented Approach to Support Software Maintenance and Reuse Activities. In Proceedings of the 5th Joint Conference on Knowledge-Based Software Engineering, JCKBSE. 2002.
25. de Vasconcelos, J.B., C. Kimble, P. Carreteiro, and Á. Rocha, The application of knowledge management to software evolution. International Journal of Information Management, 2017. 37(1, Part A): p. 1499-1506. doi: 10.1016/j.ijinfomgt.2016.05.005.
26. Serna, E.M., and A. A. Serna, Ontology for knowledge management in software maintenance. International Journal of Information Management, 2014. 34(5): p. 704-710. doi: 10.1016/j.ijinfomgt.2014.06.005.
27. Haworth, D.A., S. Sharpe, and D.P. Hale, A framework for software maintenance: A f oundation for scientific inquiry. Journal of Software Maintenance: Research and Practice, 1992. 4(2): p. 105-117. doi: 10.1002/smr.4360040204.
28. Kitchenham, B.A., G.H. Travassos, A. von Mayrhauser, F. Niessink, N.F. Schneidewind, J. Singer, S. Takada, R. Vehvilainen, and H. Yang, Towards an ontology of software maintenance. Journal of Software Maintenance: Research and Practice, 1999. 11(6): p. 365-389.
29. Anquetil, N., K.M. de Oliveira, and M.G.B. Dias, Software Maintenance Ontology. In Coral Calero, Francisco Ruiz and Mario Piattini (Eds.) Ontologies for Software Engineering and Software Technology, p 153-173. Springer Berlin Heidelberg. doi: 10.1007/3-540-34518-3_5.
30. Ulziit B., Z.A. Warraich, C. Gencel, and K. Petersen, A conceptual framework of challenges and solutions for managing global software maintenance. Journal of Software: Evolution and Process, 2015. 27(10): p. 763-792. doi: 10.1002/smr.1720.
31. Richardson, I., and C.G. von Wangenheim, Guest Editors' Introduction: Why are Small Software Organizations Different? IEEE Software, 2007. 24(1): p. 18-22. doi: 10.1109/MS.2007.12.
32. Pettersson, F., M. Ivarsson, T. Gorschek, and P. Öhman, A practitioner's guide to light weight software process assessment and improvement planning. Journal of Systems and Software, 2008. 81(6): p. 972-995. doi: 10.1016/j.jss.2007.08.032.

33. Aurum, A., F. Daneshgar, and J. Ward, Investigating Knowledge Management practices in software development organisations - An Australian experience. Information and Software Technology, 2008. 50(6): p. 511-533. doi: 10.1016/j.infsof.2007.05.005.
34. Chan, I., and C-K. Chao, Knowledge management in small and medium-sized enterprises. Communications of the ACM, 2008. 51(4): p. 83-88. doi: 10.1145/1330311.1330328.
35. Hutchinson, V., and P. Quintas, Do SMEs do knowledge management?: Or simply manage what they know?. International Small Business Journal, 2008. 26(2): p. 131-154. doi: 10.1177/0266242607086571.
36. Grimsdottir, E., and I.R. Edvardsson, Knowledge Management, Knowledge Creation, and Open Innovation in Icelandic SMEs. SAGE Open, 2018. 8(4): p. 1-13. doi: 10.1177/2158244018807320.
37. O'Connor, R., and S. Basri, Understanding the role of knowledge management in software development: a case study in very small companies. International Journal of Systems and Service-Oriented Engineering, 2014. 4(1): p. 39-52. doi: 10.4018/ijssoe.2014010103 .
38. Lin, H-F., Contextual factors affecting knowledge management diffusion in SMEs. Industrial Management & Data Systems, 2014. 114(9): p. 1415-1437. doi: 10.1108/IMDS-08-2014-0232.
39. Dingsøyr, T., N.B. Moe, and Ø. Nytrø, Augmenting Experience Reports with Lightweight Postmortem Reviews. In Proceedings of the Third International Conference on Product Focused Software Process Improvement (PROFES '01). 2001. p. 167-181. doi: 10.1007/3-540-44813-6_17.
40. Spraggon, M., and V. Bodolica, Knowledge creation processes in small innovative hi-tech firms. Management Research News, 2008. 31(11): p. 879-894. doi: 10.1108/01409170810913060.
41. Carr, M., and C. Wagner, A Study of Reasoning Processes in Software Maintenance Management, Information Technology and Management, 2002. 3(1): p. 181-203. doi: 10.1023/A:1013125112217.
42. Pino, F.J., F. Ruiz, F. García, and M. Piattini, A software maintenance methodology for small organizations: Agile_MANTEMA. Journal of Software: Evolution and Process, 2012. 24(8): p. 851-876. doi: 10.1002/smr.541.
43. Clarke, P., and R.V. O'Connor, An empirical examination of the extent of software process improvement in software SMEs. Journal of Software: Evolution and Process, 2013. 25(9): p. 981–998. doi: 10.1002/smr.1580.
44. European Commission, User guide to the SME definition. 2015. Enterprise and industry publications. Publications Office of the European Union. Luxembourg.
45. Braun, V., and V. Clarke, Using thematic analysis in psychology. Qualitative Research in Psychology, 2006. 3(2): p. 77-101. doi: 10.1191/1478088706qp063oa.
46. Creswell, J.W., Research design: Qualitative, quantitative, and mixed methods approaches, 3rd ed. 2009: SAGE Publications. Thousand Oaks, CA, USA.
47. Stojanov Z., J. Stojanov, and D. Dobrilovic, Knowledge discovery and systematization through thematic analysis in software process assessment project. In Proceedings of the IEEE 13th International symposium on intelligent systems and informatics (SISY 2015). 2015. p. 25-30. doi: 10.1109/SISY.2015.7325405.
48. Miles, M.B., A.M. Huberman, and J. Saldana, Qualitative data analysis: A methods sourcebook, 3rd ed. 2013: SAGE Publications. Thousand Oaks, CA,USA.
49. Stojanov, Z., Qualitative research on practice in small software companies. In Encyclopedia of Information Science and Technology, 3rd ed, Mehdi Khosrow-Pour (ed.). 2015. p. 650-658. Hershey, PA, USA: IGI Global. doi: 10.4018/978-1-4666-5888-2.ch062.
50. Stojanov, Z., D. Dobrilovic, and J. Stojanov, Analyzing trends for maintenance request process assessment: Empirical investigation in a very small software company. Theory and Applications of Mathematics & Computer Science, 2013. 3(2): p. 59-74.
51. Stojanov, Z., D. Dobrilovic, J. Stojanov, and V. Jevtic, Estimating Software Maintenance Effort by Analyzing Historical Data in a Very Small Software Company. Scientific Bulletin of The Politehnica University of Timioara, Transactions on Automatic Control and Computer Science, 2013. 58(2): p. 131-138.
52. Birks, M., Y. Chapman, and K. Francis, Memoing in qualitative research: Probing data and processes. Journal of Research in Nursing, 2008. 13(1): p. 68-75. doi: 10.1177/1744987107081254.
53. Corden, A., and R. Sainsbury, Exploring 'Quality': Research Participants' Perspectives on Verbatim Quotations. International Journal of Social Research Methodology, 2006. 9(2): p. 97-110. doi: 10.1080/13645570600595264.
54. Watt, D., On Becoming a Qualitative Researcher: The Value of Reflexivity. The Qualitative Report, 2007. 12(1): p. 82-101.
55. Sandelowski, M., and J. Barroso, Classifying the Findings in Qualitative Studies. Qualitative Health Research, 2003. 13(7): p. 905-923. doi: 10.1177/1049732303253488.

56. Lin, T-C., Y-S. Wang, and Y-Y. Wang, Exploring the determinants of IS developers behavioural intention to learn business skills. Behaviour & Information Technology, 2013. 32(11): p. 1125-1138. doi: 10.1080/0144929X.2013.780638.
57. Schwandt, T.A., Y.S. Lincoln, and E.G. Guba, Judging interpretations: But is it rigorous? trustworthiness and authenticity in naturalistic evaluation. New Directions for Evaluation, 2007. 114: p. 11-25. doi: 10.1002/ev.223.
58. Anney, V.N., Ensuring the Quality of the Findings of Qualitative Research: Looking at Trustworthiness Criteria. Journal of Emerging Trends in Educational Research and Policy Studies, 2014. 5(2): p. 272-281.
59. Ma, A., and B. Norwich, Triangulation and Theoretical Understanding. International Journal of Social Research Methodology, 2007. 10(3): p. 211-226. doi: 10.1080/13645570701541878.
60. Stake, R.E., Qualitative Research: Studying How Things Work. 2010: The Guilford Press. New York, USA.

AUTHOR PROFILE

Zeljko Stojanov received PhD degree in Computer science and applied informatics at the University of Novi Sad, Serbia. He works as an associate professor at the University of Novi Sad, Technical Faculty ”Mihajlo Pupin”, Serbia. He has over fifteen years of experience working with small software companies as a software developer and as a consultant in the fields of software development, software maintenance and process improvement. His research interests are in the fields of software engineering, process assessment and improvement, business informatics, learning and knowledge management, engineering education, and human aspects of engineering. He is author of scientific papers published in refereed journals and in the proceedings of international conferences. He participated in several research and industrial projects at national and international levels. He is the member of IEEE and ACM.