

Software maintenance support activities: Challenges for very small software companies

Z. Stojanov*

* University of Novi Sad, Technical Faculty "Mihajlo Pupin", Zreanjanin, Serbia

zeljko.stojanov@tfzr.rs

Abstract – Software maintenance is a set of activities performed in order to ensure proper functioning of software systems. Maintenance activities are event driven and usually are initiated by customers. Maintenance support or front-end maintenance activities are activities that provide direct contact to customers, and therefore are very important for business success of software companies. These activities in some cases require execution of back-end maintenance processes that are realized on software products. Because of specific organization of small software companies, and identified problems in their business, support activities play important role for their success in market. In this paper are outlined some challenges for very small software companies in the field of software maintenance support.

I. INTRODUCTION

Maintenance technologies supported with information technologies (IT) emerged 40 years ago [1]. Published research results revealed that maintenance has great impact on business performance such as productivity and profitability [2][3]. Implementation of effective maintenance procedures and strategies can significantly reduce problems that occur during system usage. In the context of engineering, maintenance has a wide range of meanings and includes all actions related to preserving consistency and efficiency of complex systems [4]. Cigolini et al. stated [4]: *"The scope of maintenance has grown to include not only system repair, but also system safety, economic viability, quality, and the most appropriate use of environmental resources."*

Software maintenance is generally defined as any work performed on software system that is operational and used by customers. Parikh provide the most comprehensive definition of software maintenance [5]: *"Software maintenance is the work done on a software system after it becomes operational. It includes: understanding and documenting existing systems; extending existing functions; adding new functions; finding and correcting bugs; answering questions for users and operations staff; training new systems staff; rewriting, restructuring, converting and purging software; managing the software of an operational system; and many other activities that go into running a successful software system."*

The maintenance phase of software life cycle begins after development and delivery of a software product, although some maintenance activities can be performed also during software development. Software maintenance

should be seriously considered because the maintenance costs can be several times higher than the costs during the development, especially for product with long life [6]. In literature can be found that maintenance costs are between 40% and 90% of whole costs in software life cycle. Despite that fact, software maintenance is in literature considered as the last phase in software life cycle with little attention comparing to software development. The consequence is that maintained software contains more errors than initially delivered software [7].

There are many reasons for software maintenance activities, but generally, software maintenance tasks are more complex then tasks during software development. The reasons for that are [7]:

- Documentation does not exist, or it is poor (major cause of distrust of the documentation by engineers in maintenance [8]).
- Difficulties in understanding code written by someone else.
- Programmers that develop software usually are not available to answer questions about software. Programmers are unavailable or they cannot be identified [9][10].
- Software is not designed to be easily modified.
- Maintenance is less attractive than development. Maintenance tasks are often avoided in practice [11], or are consider to be intellectually boring and tedious [12]

The main difference between software development and maintenance is that development is requirement-driven and maintenance is event-driven [13]. This means that maintenance activities are unscheduled or random events initiated mostly by customers, but also can be initiated by maintenance human resource (engineers or managers) or even by market.

Software maintenance activities are usually triggered by customers (software end users), and therefore software companies should provide efficient support services that will help in solving customer requests. The first contact with customer is handled by operative support that assists customer in submitting requests. This operative support is in practice realized as front-end support that can be independent organizational unit in a software company or is integrated into other organizational units (maintenance unit, development unit). Nevertheless, maintenance front-

end support requires more research, especially in small software companies that have specific organization and usually do not have organizational units dedicated to specific activities.

The rest of the paper is structured as follows. In the second section are outlined maintenance activities, while in the third section are outlined basic concepts of maintenance support and support activities. After that are outlined maintenance support challenges for very small software companies, and some concluding remarks.

II. SOFTWARE MAINTENANCE ACTIVITIES

Software maintenance includes short-term activities (e.g., fixing critical problems) as well as long-term activities (e.g., a major enhancement to an existing system that result in new version of existing system or in completely new software system) [14]. Nevertheless, the maintenance is generally viewed as a collection of short-term activities.

Ontology of maintenance activities is presented in [13]. Definitions of the elements in the ontology are given in Table I. Different maintenance activities require different tools and techniques, and also different skills and knowledge for maintainers. For example, different knowledge is necessary for analyzing customer problem reports and for providing training to customers. In practice, training for software system usage do not require knowledge about source code or deeper knowledge about the software architecture, while this knowledge is necessary for solving reported problems that require interventions on the software system. Further, corrective

may require a broad understanding of a large part of the product (its architecture) and various technologies used during development [8].

The most common activities in maintenance are modification activities that are, further, divided into: corrections and enhancements. Both, corrections and enhancements change software systems, but it is very important to clearly distinguish between them because they are differently observed from the financial point of view. In practice, corrections are in many cases free of charge to users, while enhancements are charged to users because they include new development and require understanding of the whole software product.

Maintenance activities can be also classified according to various levels of granularity, including the line of code level, change-log level, module level, and program level [15]. Various granularity levels require different skills and knowledge, different tools and techniques. For example, change activity at the level of the lines of code requires knowledge and skills in appropriate programming language, and usage of tools that help in finding the difference between versions of source code files (for example differences in the code can be found with *diff* utility that is integrated in many modern development environments). At the other hand, maintenance activities at module level, or program level requires additional knowledge about software architecture, requires usage of advanced modeling tools (UML modeling, conceptual modeling, XML modeling), and requires the knowledge of various technologies.

Findings of empirical study about the factors which determinate maintenance activities [16], showed that there exist predictable maintenance patterns for the business software systems. These results confirmed that usage of appropriate management and planning methods can improve various maintenance activities, and even reduce them.

Four processes that pertain to the quality of software maintenance are [17]:

- Translate customer expectations with respect to maintenance into clear service agreements.
- Use these service agreements as a basis for planning and implementing maintenance activities.
- Ensure that maintenance is done according to planning and procedures.
- Manage the internal (in software company) end external (with customers) communication about the maintenance activities carried out.

III. SOFTWARE MAINTENANCE SUPPORT ACTIVITIES

The business in many organizations is highly dependent of IT and, therefore these organizations need various kind of support during the operation and the exploitation of their IT systems. The most of the costs related to IT are IT services costs that consist of the costs of integration, operation and exploitation of IT systems [18]. These IT services include performance and

TABLE I. MAINTENANCE ACTIVITY ONTOLOGY DEFINITIONS [13]

Activity	An action of one of the following types: an investigation activity, a modification activity, a management activity, or a quality assurance activity. An activity may be made up of a number of sub-activities. Usually, it takes as input one, or more existing artifacts and outputs zero, one or many new or modified artifacts.
Investigation activity	An activity that assesses the impact of undertaking a modification arising from a change request or problem report.
Modification activity	An activity that takes one or more input artefacts and produces one or more output artefacts that, when incorporated into an existing system, change its behaviour or implementation.
Management activity	An activity related to the management of the maintenance process or to the configuration control of the maintained product.
Quality assurance activity	An activity aimed at ensuring that a modification activity does not damage the integrity of the product being maintained. Quality assurance activities may be classified as testing or certification activities.
Resource	Everything that is used to perform an activity. Resources may be hardware, software or human resources.

activities may require only the ability to locate faulty code and make local changes, while an enhancement activity

availability support, end-user and help desk support, education, and technical operation. In practice, there is still a lack of formal and structured customer orientation in IT Service Management, which is also reflected on software maintenance support services. One possible solution for this problem is usage of Service Level Agreement (SLA) as an instrument for the management of the customer's expectations [17][18]. This implies that software maintenance support can be seen as providing services to customers.

A quick and accurate response and support to customers' requests ensures their satisfaction and increases the quality of provided services. This support primarily depends on experience and skills of support personnel [19]. The beneficiaries of the efficient support are both customers and organization that provides support (customer support people and managers).

In software companies, support activities are part of front-end support, or upfront maintenance. These activities are oriented towards customers to assist them in the operation of their software and/or integrated software and hardware products [20]. This support is focused on providing immediate help to customers when they encounter problems or difficulties that prevent them from conducting their daily business.

In practice, front-end support activities vary from technical support, to participation in testing, quality auditioning, education and training, handling of emergency cases to management of customer profiles. These support activities are more or less integrated or related to the evolution and maintenance work. For example solving reported problem is closely related to evolution and maintenance, while providing training about newly integrated feature in a software product is not related to maintenance work. Each software company provides a set of support activities. The list of empirically identified types of support activities is presented in Table II [21].

The most common constellation of support levels in organizations that provide software maintenance support is presented in Figure 1. This organization is based on research reported in [20][22]. In many software companies do not exist all these levels. Each software company tailors support activities according to available resources.

At customer level are executed customer business processes supported by various software products. At this level customer identified problems or require new features in existing software products, report requests to a software company, and later helps in integrating new solutions in their business environment.

Front-end support level is primarily engaged in providing a continuous feedback to customers. The most of the working time support people spent in interactions with customers. Personnel at this level collect requests from customers, do the preliminary analyses and solve all requests that are in their domain of expertise. During the work, support personnel require help, and they find it in documentation, database, ask their colleagues (from front-end or back-end level), or even ask customers [23].

At this level are placed activities such as consultation, help desk, education and training, problem and requests elicitation, and delivery of solutions to customers. In some cases, companies at this level organize help desk [24], or software hot lines [25]. Personnel at this level have general knowledge about software and required IT infrastructure for its proper functioning. More complex problems or requests that require deeper understanding of software product are forwarded to back-end support level.

TABLE II. TYPES OF FRONT-END SUPPORT ACTIVITIES [21]

Software problem management
Testing
Participate in change management process
Quality Control
Consultation
Technical Support
Education and Training of customers
Self-education and training
Preventive maintenance
Passive Requirements Elicitation
Active Requirements Elicitation
Monitoring the process
Measure customer satisfaction
Managing emergency cases
Managing disaster recovery
Manage customer profile

Knowledge is crucial for software support, because solving specific problems require understanding of specific details relevant for that problem. These relevant details come in many different kinds [25]: details about software product (features, known bugs, etc.), details about the hardware (disk drives, monitors, networks, etc.), details about compatibility with other software (business information systems, operating systems, telecommunications, etc.) and so on. Further knowledge that is necessary part of software support is related to details about customer's business and their hardware and

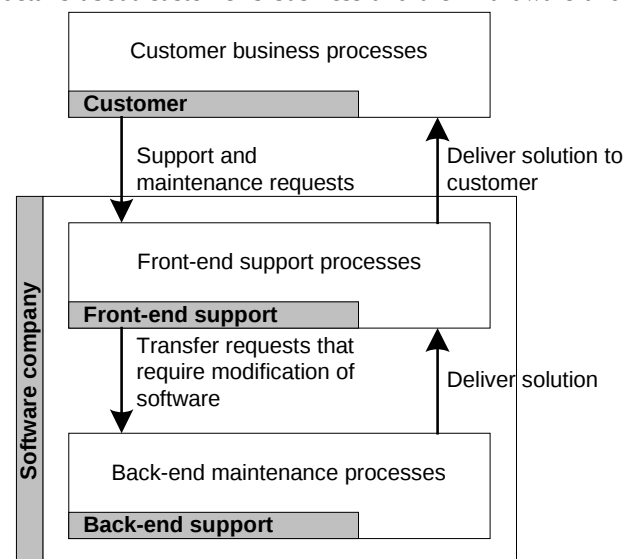


Figure 1. Organization of support levels in software companies

software infrastructure. All this details are important for proper interpretation of problems identified by customer.

Mastery of technical details is not enough for effective software support [25]. Social skills are also very important for personnel in software support. They should possess pragmatic reasoning in all situations, they must be able to discern customers' needs and hopefully, meet them, and they must be able to put some distance between themselves and their work and not take things too personally. Typical situation when social skills are very important are when user makes "stupid mistake" and report serious problem with software, or when user that reports problem is angry.

Software front-end support requires service from back-end support when support personnel cannot solve technical problem or request reported by customers. In such a situation, they transfer confirmed problem to maintenance engineers that are capable to solve stated problem (see Figure 1). The activities at back-end are problem investigation, identification of problem causes (from technical point of view), problem resolving, testing and delivery of solutions. At this level are performed typical software maintenance and evolution tasks related to modification of software products.

Feedback between persons on all levels presented at Figure 1 is important for proper and efficient functioning of support. In the context of software maintenance support that is composed from social and technical elements, the term feedback have the meaning like in management psychology and human factors research. Feedback relates to the presentation of information to individuals regarding different aspects of performance, such as behavior, strategies, or outcomes [26]. Feedback has many dimensions such as source, timing, specificity, type, and frequency. Feedback provides information on how well and effective individuals are performing relative to their goals, how effective specific behaviors are for a given task, and also suggests how much task effort is required to achieve specific objectives [27].

IV. CHALLENGES FOR VERY SMALL SOFTWARE COMPANIES

Small companies are dominant in economies across the globe [28]. U.S. Census Bureau's "1995 County Business Patterns" pointed that the vast majority of software and data processing companies are small, and that those with more than 50 employees comprise only a few percent of the total number [29]. Laporte et al. reported in [30] that in Europe, 85% of IT sector companies have between 1 and 10 employees, while in Montréal area in Canada over 50% have fewer than 10 employees. Organizations that have less than 25 employees are defined as very small enterprises (VSEs) [30][31][32]. The common situation in the majority of software VSEs is that many individuals are responsible for multiple roles (developer, architect, manager and team leader) [33]. Because of many unexpected problems, improvisation is common in the practice [34]. Comprehensive breakdown of the proposed infrastructure and support processes for software VSEs is presented in [32]. This breakdown includes all activities during the whole life cycle of software.

The main challenges posed to software VSEs are [28]: managing and improving their software processes, dealing with rapid technology advances, maintaining their products, operating in a global software environment, and sustaining their organizations through growth. Empirical evidence presented in [35] suggests that software VSEs should combine high level of customer orientation, with high levels of entrepreneurial and technology orientations. Hakala and Kohtamäki reported that 67% of software companies provide various maintenance or user support services.

Challenges identified in [28] can be applied also to software maintenance support activities. Because of the importance and great cost of maintenance activities, there is a need to investigate them in more details. Regarding identified problems within front-end support in [20] the following challenges for software VSEs should be investigated: organizational, process, human factors, and product challenges.

A. Organizational challenges

Organizational challenges are related to the size and the complexity of a software company that put restrictions on organizing effective support. The questions are:

- Do software VSEs have enough *resources* (human and technical) to provide effective support to their customers?
- What *expertise* is needed and how to acquire it in order to provide effective support? This aspect is concerned with knowledge, skills, and experience of software maintenance personnel.
- What other *barriers* (social, administrative, political, and cultural) companies meet in organizing and providing support?
- *Outsourcing*. How company deals with problems that arise during outsourcing products and services. This is very important because some very small companies do outsourcing to larger companies worldwide.

B. Process challenges

Process challenges are related to defining and establishing effective support processes. The directions for investigation are:

- *Definition of processes and problems of adherence*. This includes investigations which processes exist, are they formalized and agreed with standards or company business policies, and if there is any automated support for processes.
- *Communication within a software company*. This includes communication between people, between teams or departments, and communication with other software companies that deliver third party software components or services.
- *Customer satisfaction*. This includes: how software company meets customer expectations (realistic and unrealistic), does customer orientation increases business of both software

company and customer, does company provide continuous feedback to customers (demands, requests, products), contracts and service level agreements. Active involvement of users in maintenance is important and can positively affect customer satisfaction [36].

C. Challenges related to human factors

Software engineering is human activity with different roles (engineers, designers, managers, users) and behaviors (motivation, communication, understanding) of involved people. Challenges in software companies are related to shortage with maintenance personnel, and their competences and attitude towards support tasks (in many cases indifferent or detached attitude), while at customer side challenges are related to their knowledge about the used software products, and proper usage of software products. Investigation of these, human or people factors requires usage of sociological and psychological aspects of human behaviors [37][38].

D. Product challenges

Many attributes of software products affect maintenance support activities. These attributes are product quality, product complexity, maintainability, user-friendliness, etc. It is challenging to investigate how these attributes correlate, and how they affect processes, and all people involved in maintenance support activities.

V. CONCLUSIONS

In this paper are presented challenges that software companies face in the field of maintenance support towards their clients. These challenges are not only technical or technological, but are also related to organization and management, social and cultural context. The main characteristics of software VSEs have been presented in available literature (see section IV of this paper), and the question is: *How software VSEs meet presented challenges in the field of maintenance support?*

Although many international standards (for example *ISO 29110 Software process lifecycles for very small entities*) have been recently developed and adapted to software VSEs [39], the most of small software companies do not accept them nor plan to do that [40]. Survey of existing literature provides the evidence that there is a limited amount of empirical research investigating the support activities in software VSEs.

Further research should be directed towards assessment of current practice, and development and implementation of appropriate tools and methods that will improve the practice. One of the main problems in small software companies is the lack of time and resources to invest in the research. This problem can be overcome through cooperation with universities, or joint development and service providing by more small companies.

ACKNOWLEDGMENT

This research is financially supported by Ministry of Science and Technological Development, Republic of

Serbia under the project number TR32044 "The development of software tools for business process analysis and improvement", 2011-2014.

REFERENCES

- [1] M. Kans, "The advancement of maintenance information technology: A literature review", *Journal of Quality in Maintenance Engineering*, vol. 5, no. 1, 2009, pp. 5-16, DOI: 10.1108/13552510910943859.
- [2] I. Alsyouf, "The role of maintenance in improving companies' productivity and profitability", *International Journal of Production Economics*, vol. 105, no. 1, 2007, pp. 70-78, DOI: 10.1016/j.ijpe.2004.06.057.
- [3] M. Kans and A. Ingwald, "Common database for cost-effective improvement of maintenance performance", *International Journal of Production Economics*, vol. 113, no. 2, 2008, pp. 734-747, DOI: 10.1016/j.ijpe.2007.10.008.
- [4] R. D. Cigolini, A. V. Deshmukh, L. Fedele, and S. A. McComb, editors. *Recent Advances in Maintenance and Infrastructure Management*, Springer London, London, UK, 2009. DOI: 10.1007/978-1-84882-489-8.
- [5] G. Parikh, "Exploring the world of software maintenance: what is software maintenance?", *ACM SIGSOFT Software Engineering Notes*, vol. 1, no. 2, April 1986, pp. 49-52, DOI: 10.1145/382248.382820.
- [6] I. Sommerville, *Software engineering*, 6th edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
- [7] S. Mamone, "The IEEE standard for software maintenance", *ACM SIGSOFT Software Engineering Notes*, vol. 19, no. 1, 1994, pp. 75-76, DOI: 10.1145/181610.181623.
- [8] J. Singer, "Practices of Software Maintenance", In *Proceedings of the International Conference on Software Maintenance, ICSM '98*, pp. 139-, Washington, DC, USA, 1998.
- [9] J. Connell and L. Brice, "Prolonging the life of software", In *Proceedings of the July 9-12, 1984, national computer conference and exposition, AFIPS '84*, pp. 243-249, New York, USA, 1984, ACM, DOI: 10.1145/1499310.1499342.
- [10] E. Arisholm, L. C. Briand, S. E. Hove and Y. Labiche, "The Impact of UML Documentation on Software Maintenance: An Experimental Evaluation", *IEEE Transactions on Software Engineering*, vol. 32, no. 6, June 2006, pp. 365-381, DOI: 10.1109/TSE.2006.59.
- [11] P. Stachour and D. Collier-Brown, "You Don't Know Jack About Software Maintenance", *Communications of the ACM*, vol. 52, no. 11, 2009, pp. 54-58, DOI: 10.1145/1592761.1592777.
- [12] R. Mookerjee, "Maintaining enterprise software applications", *Communications of the ACM*, vol. 48, no. 11, pp. 75-79, November 2005, DOI: 10.1145/1096000.1096008.
- [13] B. A. Kitchenham, 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*, vol. 11, no. 6, 1999, pp. 365-389.
- [14] R. D. Banker and S. A. Slaughter, "A field study of scale economies in software maintenance", *Management Science*, vol. 43, no. 12, *Frontier Research on Information Systems and Economics*, 1997, pp. 1709-1725, DOI: 10.1287/mnsc.43.12.1709.
- [15] S. R. Schach, B. Jin, L. Yu, G. Z. Heller and J. Offutt, "Determining the Distribution of Maintenance Categories: Survey versus Measurement", *Empirical Software Engineering*, vol. 8, no. 4, 2003, 351-365, DOI: 10.1023/A:1025368318006.
- [16] C. F. Kemerer and S. A. Slaughter, "Determinants of software maintenance profiles: an empirical investigation", *Journal of Software Maintenance*, vol. 9, no. 4, July 1997, pp.235-251. DOI: 10.1002/(SICI)1096-908X(199707/08)9:4<235::AID-SMR153>3.0.CO;2-3.
- [17] F. Niessink and H. van Vliet, "Software maintenance from a service perspective", *Journal of Software Maintenance: Research and Practice*, vol. 12, issue 2, March/April 2000, pp. 103-120.

- [18] J. Bouman, J. Trienekens and M. Van der Zwan, "Specification of Service Level Agreements, Clarifying Concepts on the Basis of Practical Research", In *Proceedings of the Software Technology and Engineering Practice (STEP '99)*, 169-178. DOI: 10.1109/STEP.1999.798790.
- [19] S. Heras, J. A. Garcia-Pardo, R. Ramos-Garijo, A. Palomares, V. Botti, M. Rebollo, and V. Julian, "Multi-domain case-based module for customer support", *Expert Systems with Applications*, vol. 36, issue 3, part 2, April 2009, pp. 6866-6873, DOI: 10.1016/j.eswa.2008.08.003.
- [20] M. Kajko-Mattsson, "Problems within front-end support", *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 16, issue 4-5, 2004, pp. 309-329, DOI: 10.1002/smr.298.
- [21] M. Kajko-Mattsson, "Taxonomy Of Front-End Support Activities", *Journal of Integrated Design & Process Science*, vol. 8, issue 1, January 2004, 19-29.
- [22] M. Kajko-Mattsson, "Infrastructures of Virtual IT Enterprises", In *Proceedings of the International Conference on Software Maintenance 2003 (ICSM '03)*, 199-208, DOI: 10.1109/ICSM.2003.1235422.
- [23] B. T. Pentland, "Organizing Moves in Software Support Hot Lines", *Administrative Science Quarterly*, vol. 37, no. 4, December 1992, pp. 527-548.
- [24] M. Kriegsman and R. Barletta, "Building a Case-Based Help Desk Application", *IEEE Expert: Intelligent Systems and Their Applications*, vol. 8, no. 6, December 1993, pp. 18-26, DOI: 10.1109/64.248349.
- [25] B. T. Pentland, "Bleeding Edge Epistemology: Practical Problem Solving in Software Support Hot Lines", In Stephen R. Barley and Julian Edgerton Orr, editors, *Between craft and science: technical work in U.S. settings*. Ithaca, NY, USA, ILR Press, 1997.
- [26] P. W. B. Atkins, R. E. Wood and P. J. Rutgers, "The effects of feedback format on dynamic decision making", *Organizational Behavior and Human Decision Processes*, vol. 88, issue 2, July 2002, pp. DOI: 10.1016/S0749-5978(02)00002-X.
- [27] C. F. Lam, D. S. DeRue, E. P. Karam and J. R. Hollenbeck, "The impact of feedback frequency on learning and task performance: Challenging the 'more is better' assumption", *Organizational Behavior and Human Decision Processes*, In Press, Corrected Proof, DOI: 10.1016/j.obhdp.2011.05.002.
- [28] I. Richardson and C. G. von Wangenheim, "Guest Editors' Introduction: Why are Small Software Organizations Different?", *IEEE Software*, vol. 24, No. 1, 2007, pp. 18-22, DOI: 10.1109/MS.2007.12.
- [29] M. E. Fayad, M. Laitinen and R. P. Ward, "Thinking objectively: software engineering in the small", *Communications of the ACM*, vol. 43, issue 3, March 2000, pp. 115-118, DOI: 10.1145/330534.330555.
- [30] C.Y. Laporte, A. Renault, S. Alexandre and T. Uthayanaka, "The Application of ISO/IEC JTC 1/SC7 Software Engineering Standards in Very Small Enterprises", *ISO Focus*, International Organisation for Standardisation, September 2006, pp 36-38.
- [31] N. Habra, S. Alexandre, J-M. Desharnais, C. Y. Laporte and A. Renault, "Initiating software process improvement in very small enterprises: Experience with a light assessment tool", *Information and Software Technology*, vol. 50, issues 7-8, June 2008, pp. 763-771, DOI: 10.1016/j.infsof.2007.08.004.
- [32] V. Ribaud, P. Saliou, R. V. O'Connor and C. Y. Laporte, "Software Engineering Support Activities for Very Small Entities", *Systems, Software and Services Process Improvement, Communications in Computer and Information Science*, 2010, vol. 99, pp. 165-176, DOI: 10.1007/978-3-642-15666-3_1.
- [33] A. Rainer, "The Value of Empirical Evidence for Practitioners and Researchers", *Empirical Software Engineering Issues. Critical Assessment and Future Directions, Lecture Notes in Computer Science LNCS 4336*, pp. 24, 2007, DOI: 10.1007/978-3-540-71301-2_8.
- [34] T. Dyba, "Improvisation in Small Software Organizations". *IEEE Software*, vol. 17, no. 5, pp. 82-87, DOI=10.1109/52.877872.
- [35] H. Hakala, M. Kohtamäki, "Configurations of entrepreneurial-customer- and technology orientation: Differences in learning and performance of software companies", *International Journal of Entrepreneurial Behaviour & Research*, vol. 17, issue 1, pp. 64 - 81, DOI: 10.1108/13552551111107516.
- [36] P. Bhatt, G. Shroff and A. K. Misra, "Dynamics of software maintenance", *ACM SIGSOFT Software Engineering Notes*, vol. 29, issue 5, September 2004, pp. 1-5, DOI=10.1145/1022494.1022513.
- [37] H. Sharp and H. Robinson, "Some social factors of software engineering: the maverick, community and technical practices", *ACM SIGSOFT Software Engineering Notes*, vol. 30, issue 4, July 2005, pp. 1-6. DOI=10.1145/1082983.1083117.
- [38] J. Sillito and E. Wynn, "The Social Context of Software Maintenance", In *Proceedings of the International Conference on Software Maintenance, icsm 2007*, pp. 325-334, DOI: 10.1109/ICSM.2007.4362645.
- [39] C. Y. Laporte, S. Alexandre and A. Renault, "Developing International Standards for Very Small Enterprises", *Computer*, vol. 41, no. 3, pp. 98-101. DOI: 10.1109/MC.2008.86.
- [40] S. Basri and R. V. O'Connor, "Understanding the Perception of Very Small Software Companies towards the Adoption of Process Standards", *Systems, Software and Services Process Improvement, Communications in Computer and Information Science*, 2010, Volume 99, pp. 153-164, DOI: 10.1007/978-3-642-15666-3_14.