The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.99 e-ISSN: 2251-7545

A Methodology for Product Development based on Open Source Software

Majid Tajamolian
Department of Computer Eng., Taft Branch, Islamic Azad University, Taft, Iran
tajamolian@taftiau.ac.ir

Majid Taghiloo
Pishgaman Kaipod Kavir Co., R&D Center
taghiloo@kaipod.ir

Mohammad Ali Agheli
Pishgaman Kaipod Kavir Co., R&D Center
agheli@kaipod.ir

***Abstract*-- Traditional software development methods have different phases that must be accomplished step by step. The most important stage is the software analyzing and design phase that the result architecture will be the base of implementation. Since the framework of software is created from scratch, maximum flexibility can be found in the architecture design and development of software. But In term of methodology, product development based on open source software is different from traditional methods. In this method, software product will be produced by integration of the separate open source modules. Each of these modules is an independent standalone product and to cover the additional functional requirements, they must be putted together. To provide its own functionalities, each independent module uses a set of Blocks as a architectural component of module. In this paper, a new methodology is proposed to describe all of the challenges in the course of product development based on open source software.**

Keywords: Software development methodology; Open source; FOSS; Security;

I. INTRODUCTION

The amount of free and open source software has increased exponentially along the expansion of the Internet, and currently there are at least hundreds of thousands of such software projects, though majority of them are very small. It has a good potential in rapid and robust development of software products.

Open source software development represents a fundamentally new concept in the field of software engineering. Open source development and delivery occurs over the Internet. Developers are not confined to a geographic area. They work voluntarily on a project of their choice. As new requirements emerge, the software is enhanced by the user/developers. This paper explores patterns and processes that use in product development with open source projects. FOSSD is a relatively new way of building and deploying large software systems on a global basis, and differs in many interesting ways from the principles and practices traditionally advocated for software engineering [1].

According to its proponents, open source style software development has the capacity to compete successfully, and perhaps in many cases displace, traditional commercial development methods. In order to investigate such claims, there are examined data from two major open source projects, the Apache web server and the Mozilla browser in J. Herbsleb article [2]. As a result of previous researches, the highly efficient bug fixing process and quick release cycles are considered key properties of the open source software development methodology [3, 4 and 5].

The concept of Free ("Free" as "Free Speech" not as "Free Drink"!) and Open Source Software (FOSS) is not new. It has been around since the 1960s in universities such as MIT and corporate firms such as Bell Labs who freely used source code for research [6, 7, 8 and 9].

But Open Source Software Development is a recent phenomenon, while traditional closed source software development has been here since the dawn of software development. The basic tenets of open source project development are clear enough. For example, one major difference between these two models is source code visibility. It is useful for us to avoid any reworks. According to its proponents, open source style software development has the capacity to compete successfully, and perhaps in many cases displace, traditional commercial development methods.

In this paper, we briefly describe the open source software development and propose a new approach for product development based on open source projects. The paper is organized as follows; in section 2, we describe basic concepts of proposed methodology. In section 3, we describe the detailed development processes of the methodology by presenting the activity diagram of each step. Finally, the life cycle model for open source based development of product will be proposed.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.99 e-ISSN: 2251-7545

II. PRODUCT DEVELOPMENT METHODOLOGY BASED ON FOSS

The architecture and interface are the two important attributes of the all proposed solutions. Architectural components of this methodology are MODULEs. A module is software with one or more correlated BLOCKs that connected by its interfaces. The interfaces of the module are the total interfaces of its blocks. The module title is used for open source software with the following properties:

1. Standard: using it in the module mode is possible when its blocks interfaces, documentations, the way of combination and deployment are predefined.
2. Independent: In the absence of particular service need, interaction between blocks must be occurred through the specific interfaces (By the method of inter-block communication).
3. Well-documented: Documentation of the module must be complete and then it can easily be used in other products.

In our case study, the new XTM product that is belonging to KAIPOD [10] is developed based on this methodology. It is made by different modules that each of them contains one or more blocks (open source or commercial). The block is the smallest part of the product architecture. The communication between blocks requires clear definition of interfaces. There are several ways to provide the interface:

1. Inter Process Communication
2. Network (Loop Back)
3. Supervisor Call (Socket)
4. API

Hence, interface specification must be defined clearly in the module document and module model. It defines the services that are providing by the module to the other module and it should define the services that specifies what services must be made available for the module to execute as specified. A module model is a definition of standards for module implementation, documentation and deployment. It specifies how interfaces should be defined and the elements that should be included in an interface definition. The following list is the output of the module model:

1. Extract the module blocks list
2. Mapping the blocks to the available open source software or the new software piece that should be developed.
3. Specify the blocks inter-connection and specification of their interfaces

Briefly, the module model is the set of standards that during the product development must be followed.

III. PRODUCT DEVELOPMENT PROCESS

Software development based on proposed methodology will be accomplished on some different separated units that entitled by: System Analyzing unit, System Development unit, System Integration unit, System Testing unit, and System Maintenance unit. The system analyzing unit is the main part of the methodology and it has a role like a heart in the body system. All of the diagrams in the next sections are followed by this unit of the methodology. System development unit will implement all required new blocks and also it is responsible for customizing of the existing blocks per case. A Module will be produced based on the integrating of all corresponding Blocks. The similar scenario is required to produce a product from its corresponding modules. All of these tasks will be managed by System Integration group. Finally, System Testing and Maintenance units are responsible for correctness and supporting of the final product.

In addition to the mentioned units, there are some entities that play important roles in the methodology: End users, Documents patterns, Product backlogs, Module backlogs, Block Backlogs, Project Baseline and Documents. End user is an important entity of the methodology. Its requirements must be applied to the final product. Unified structure of documentation is useful for information sharing of different units of the product development team. In order to gradual development of product, similar to the agile methodologies, the product is divided to the small parts (modules); the name of this list is "product backlog". Each module can be a separate complex system. The same method for gradual development of module and block will be applied. Finally, to prepare a unique glossary and dictionary of project, the project baseline will be useful.

The process flow diagram is used to demonstrate the all development processes of the different phases of the proposed methodology. It includes the information regarding the connection between various units and also consists of the process piping and major parts details.

A. System requirement analyzing and design process

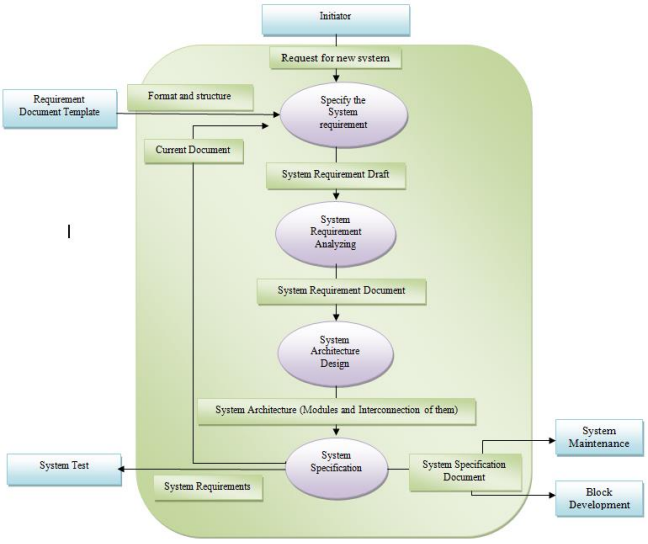

Fig. 1. system requirement analyzing and design diagram

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.99 e-ISSN: 2251-7545

First of all you need to understand the needs of your client, the one who made the request for the product. Requirements are the most basic understanding about the business logics of the problem to be solved. The aim of the first phase of the methodology is taking notes of details and maintaining a list of the functionalities and behaviors of the product to be developed. This is the first and main step of methodology is when we gather the System Requirements, or all the details of the Business Logic of the software, to be translated later to a working Software. According to the needs and goals of the product, system specification document is provided based on the system requirement template document. This document to be used by different parts of projects: System Maintenance, System development, and system testing units.

The process flow diagram of figure 1 includes the information regarding the connection between various units and also consists of the process piping and major parts details.

B. Module requirement analyzing and design process

The open source software that is used in final product architecture called Module. Determining the requirement specification of the system is essential to select the best open source software for corresponding module. It helps the appropriate and consistent solution with the system architecture requirement to be chosen. For this purpose, as shown in Figure 2, after gathering requirements of the module, internal structure and architecture of the module are designed and then the result architecture specification document will be delivered to the system maintenance, block development and system testing units, as a reference document.

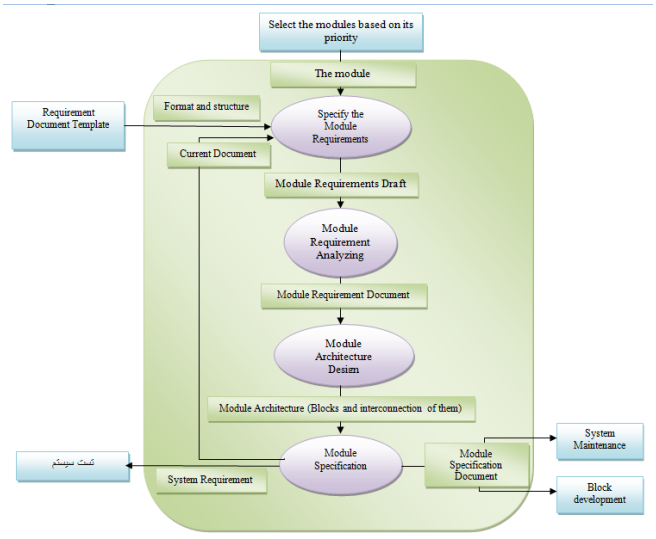

Fig. 2. Module requirement analyzing and design diagram

C. Block requirement analyzing process

Block is the smallest part of the software product's structure in the proposed methodology. The Module is created based on combining of related blocks. In this stage of the product development process, all required blocks will be identified to create the corresponding modules and for each of these blocks, specification of the requirement will be extracted in the format of document template.

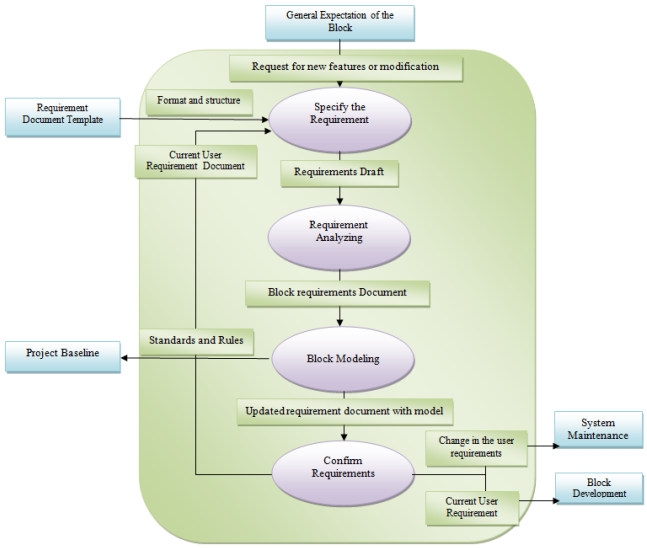

Fig. 3. Block requirement analyzing diagram

After specifying the requirements, the result draft document is used to create architecture of the block. The next step is creating a block model which it contains detailed information of the architecture like: interconnection interfaces with other blocks, inputs, outputs, and the details of the processing behaviors. The final requirement specification document of the block with its model will be delivered to the maintenance and development phases of the methodology as reference documents.

D. Block Implementation process

Now this is the time of implementing the smalest part of our product with the name of BLOCK. In this phase, the programmers group can start the implementation phase based on the block requirement analyzing document. In consideration of the fact that in this methodology, the main aim is maximum usage of the previous available open soure works, the implementation process can be done based on the following two separate paths:

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.99 e-ISSN: 2251-7545

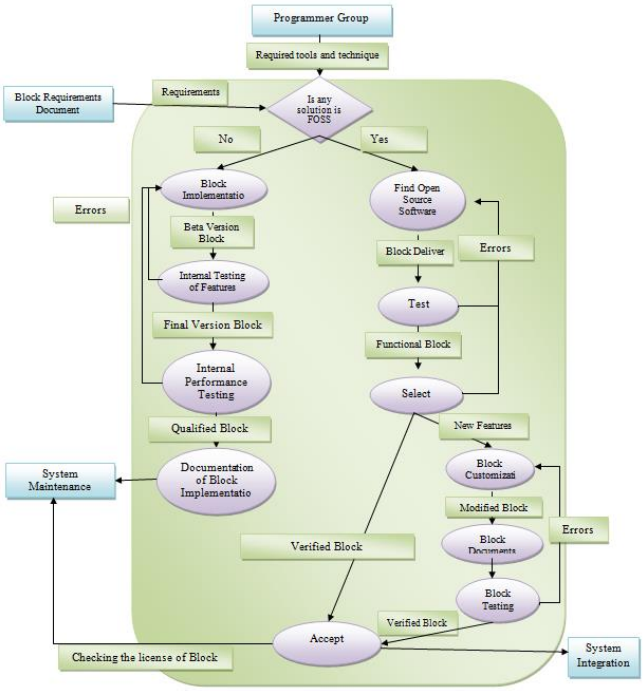

Fig. 4. Block implementation diagram

In order to implement a block, first of all the proper (open source) software candidate with the maximum coverage of the functional requirements will be selected to do the complementry testing. In the case of existance of more than one condidate for the corresponding block, the solution with more active supporting community will be selected to passing the complementary tests. After that in the required case of adding new features it should be customized. The block customization document, that is useful for future modification and maintenance will be generated. After final testing of the block, if it is error-free, it will be delivered to the system integration phase and also the produced document will be useful for the system maintenance phase. If for the corresponding block, there is no available open source alternative, it should be implemented from scratch. In this way, after finishing the implementation, functional and performance capability of the block will be tested and by getting the successful results, it can be used in the product. Implementation document will be delivered to the maintenance phase.

E. Module integration process

After finalizing the block implementation process, this is the time for integrating the blocks and creating the unified module. To do this, the integration process will be started based on the module architecture (that was the result of the module requirement analysing phase). After getting a successful result from internal testing, the final modules documents will be delivered to the product integration and maintenance phases. In the case of the failure occurrence in the testing phase of the modules, if it is an open source solution, the corresponding block must be replaced with its alternative. Otherwise the new requirement document must be sent to the module development section. Then with the new block the problem will be resolved. Module document for continuity of the system life cycle will be delivered to the maintenance and integration phase.

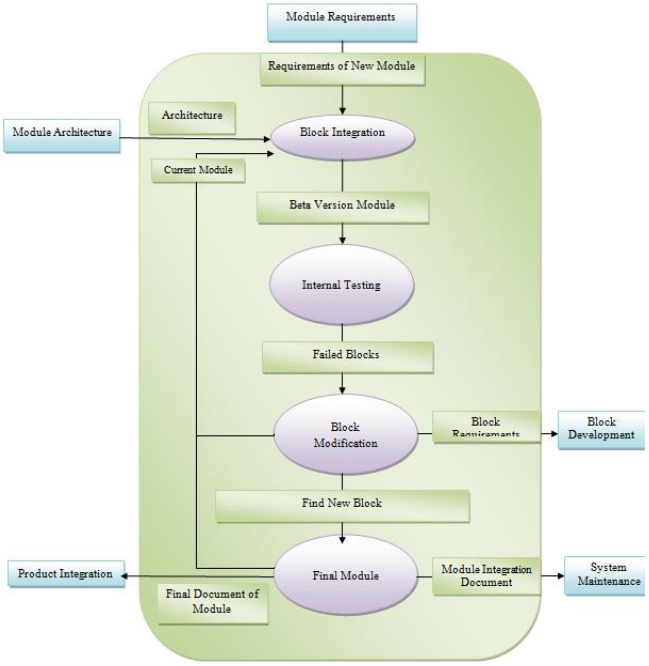

Fig. 5. Module integration diagram

F. Product integration process

After finishing the previous stages successfully, the process of product integration is not hard. In this phase, based on the system requirements of the product and by referring to the product architecture that is produced in previous stage, the integration process will be started. After arranging the modules with together and unified configuring the system, the beta version of the product will be integrated. It needs to pass some tests. In the case of failure, the failed module(s) must be corrected or in the worst case it/they must be regenerated.

Otherwise, final system is ready to use and integration documents will be sent to the maintenance phase. Also, this document will be hold in testing phase for future referring.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.99 e-ISSN: 2251-7545

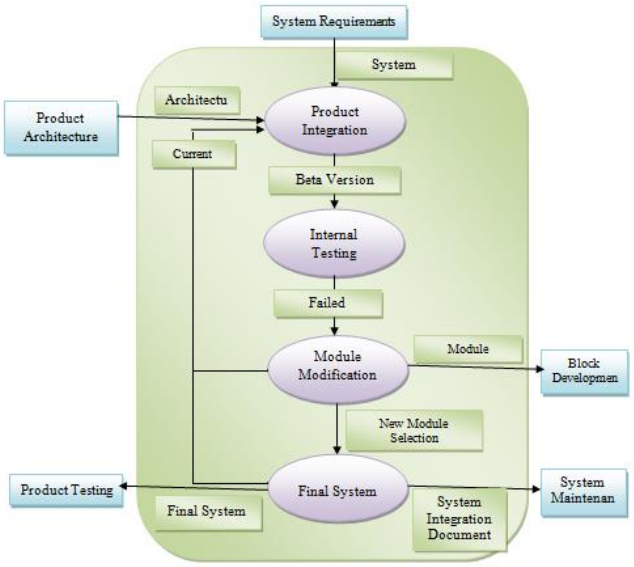

Fig. 6. Product integration diagram

G. *Testing process*

Testing phase is one of the important stages of product development and it has a good affect in the business continuity of product. Therefore based on the previous product experiences, the document of testing strategy will be created. After that, based on the system testing plan and referring to the documents of the module implementation and integration phases, the test processes will be executed. If the all tests in the aspect of end user requirements have been passed, the product can be released with high assurance.

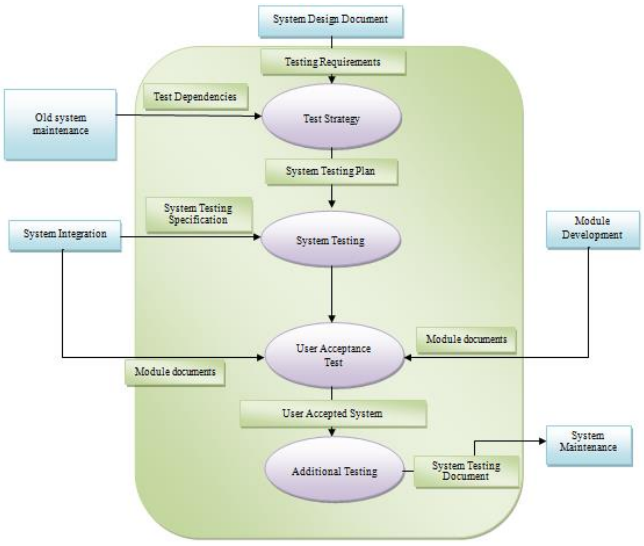

Fig. 7. Testing diagram

H. *Maintenance process*

All of the result documents will be used in this phase and in the case of users' error reporting, the maintenance phase will start its work based on its strategies.

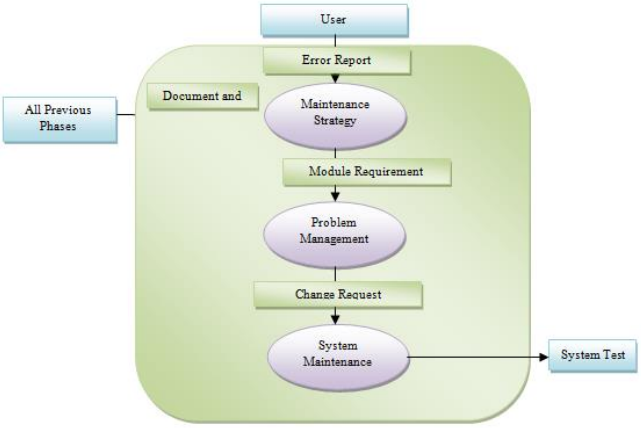

Fig. 8. Maintenance Diagram

IV. CONCLUSION

A highly efficient bug fixing process and quick release cycles are considered key properties of the open source software development methodology. This paper proposes a new methodology of product development based on open source solutions. This new approach is the result of our experience in the real world. Our studies, discussions and experiences during a real project, have shown that traditional methodologies of software development are not so suitable for the product development based on the open source solutions. The working processes in requirement analyzing, designing and implementing are deferent from the traditional methods. In this paper, by describing the eight diagrams, the working processes of the new approach for the different phases of the product life cycle, have been shown. This, however, is only a starting point for this kind of product development. In conclusion, the main aim in the open source centric products is rapid development (based on reuse of the other open source software) of robust product in a reduced cost. Therefore, we believe that new methodology is required to resolve its specific challenges. The more detailed information of this approach in demonstrating the roles of agile methods is our future work.

ACKNOWLEDGMENT

Majid Tajamolian thanks Islamic Azad University (Taft Branch) for financial support of this paper publication. Also all of the authors want to offer special thanks to Pishgaman Kaipod® Kavir Co. for its great support & sponsorship of the XTM project that yields this paper.

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.99 e-ISSN: 2251-7545

REFERENCES

[1] W. Scacchi, J. Feller, B. Fitzgerald, S. Hissam and K. Lakhani, "Understanding Free/Open Source Software Development Processes, SOFTWARE PROCESS IMPROVEMENT AND PRACTICE", pp.95–105, 2006.

[2] A. MOCKUS, R. T. FIELDING, and J. HERBSLEB, "Two Case Studies of Open Source Software Development: Apache and Mozilla"

[3] Challet, D., Du, Y.L.: "Closed Source versus Open Source in a Model of Software Bug Dynamics", Cond-Mat/0306511 (June 2003), http://arxiv.org/pdf/cond-mat/0306511

[4] Feller, J., Fitzgerald, B.: "Understanding Open Source Software Development". Addison-Wesley Professional, Reading (2001)

[5] Weinstock, C.B., Hissam, S.A.: "Making Lightning Strike Twice? " In: Feller, J., Fitzgerald, B., Hissam, S., Lakhani, K. (eds.) "Perspectives on Free and Open Source Software", pp. 93–106. MIT Press, Cambridge (2005).

[6] Steven Webber, ."The Political Economy of Open Source Software.", California, June 2000.

[7] Bruce Perens, ."The Open Source Definition." Available at: http://perens.com/Articles OSD.html Acc. on: Oct 2002

[8] Malcolm M. ."Profit Motive Splits Open Source Movement.", Aug 26th, 1998. Available at http://content.techweb.com/wire/ story/TWB19980824S0012 Acc. on: Oct 2002

[9] "What is Free Software Foundation?" Available at: http://www.gnu.org/fsf/fsf.html Acc. on: Oct 2002

[10] http://www.kaipod.ir

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.99 e-ISSN: 2251-7545

Authors

[1] Majid Tajamolian is currently a lecturer in Islamic Azad University (Taft Branch). He is also a well-known IT consultant through Yazd province, central of Iran. He has been a senior member of Pishgaman Kaipod® Kavir * R&D Center during the last 2 years. He completed his MSc degree in Computer Software Engineering, in 1999 at Amirkabir University of Technology. He did his BSc in Computer Software Engineering, in 1994 at Isfahan University of Technology. His areas of interests include Free Open Source Software (FOSS), IT Security, e-banking, IT Strategic Planning & Management, and Hybrid Networks Management. (tajamolian@taftiau.ac.ir)

[2] Majid Taghiloo is currently a researcher in Pishgaman Kaipod® Kavir * Co. He completed his MSc degree in Information Technology Engineering, in 2008, at Amirkabir University of Technology. He did his BSc in Computer Software Engineering, in 2004 at Tehran University of IRAN. His areas of interests include Security, Ad Hoc Network, Information Technology and E-Commerce.
(taghiloo@kaipod.ir)

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco State University, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.99 e-ISSN: 2251-7545

[3] Mohammad Ali Agheli is currently CEO of Pishgaman Kaipod® Kavir [*] Co. He completed his MSc degree in Information Technology Eng. in 2011, at Amirkabir University of Technology. He did his BSc in Computer Software Engineering, in 2002 at Islamic Azad University (Meybod branch). He is also top manager of Firewall Projects R&D activities in Pishgaman Kavir Group.
(agheli@kaipod.ir)

[*] Pishgaman Kaipod® Kavir is an Iranian company in the information security field. The company is a member of Pishgaman Kavir Group which is the biggest cooperative company in Iran in the ICT field. (http://www.kaipod.ir)