Inm- An Integration Model Resolving 3rd Party Source Code/Component Incorporation Issues During Open Source Software Development

Mariam Nosheen and Zahwa Syed
Lahore College for Women University, Pakistan

Abstract: *In Open Source Software Development (OSSD) Community, numbers of designers collaborate with each other for developing software by using or modifying source codes of previously developed software. OSS is well known for free software which can be used by designers with freedoms on cost and copyrights. Many open source designers need to integrate third party open source codes and components within their target code. While integration, the designers might face various OSS issues like Compatibility, Interoperability, Testability and Documentation. The designers might also face some Performance and Privacy related issues after source code integration. In order to overcome all these issues, in this paper, a model named InM (Integration Model) has been proposed which will enable the designers to tackle all the integration issues in a manageable and manner able way. At the end, the proposed model is verified by taking an open source case study named "Moodle". In order to overcome the Performance and Privacy issue that arises after integration, an integrated source code has been monitored on a weekly basis and the monitoring details are shown in the form of statistics.*

Keywords: *Open Source Software, Integration Model, Third Party Source Code, Open Source Integration.*

Received November 16, 2012; Accepted March 13, 2013

1. Introduction

Open Source Software Development (OSSD) is becoming a viable software development approach that is being used by many software users and development companies. OSSD projects are basically internet-based communities of software designers who voluntarily collaborate to develop or modify the software that they or their organizations need [15]. Open source software projects and development processes have spread rapidly and widely. The number of designers participating in each OS project ranges from a few to many thousands, and the number of users of the Open Source Software (OSS) produced by open source software development projects range from few to many millions [22]. OSSs are mainly developed by the programmers who freely participate, distribute source code in geographically distributed environment, communicating over the internet [19, 5]. The initiative behind open source is *"software that must be distributed under a license that guarantees the right to read, redistribute, modify and use the software freely"* [25].

When a developer starts developing new project or software, there might be certain circumstances when they need to integrate other developer's OSS source codes or components within their source code. Any developer either working on open source or closed source software projects might face such circumstances. While integrating different source codes, the designers might face various issues. This is because the source code and components which the designers needs to integrate within his/her source code might be developed using different approaches and platforms as all the designers have their own state of mind and they think differently. It is very rare when the thinking of multiple designers matches with each other. So when the new developer incorporate third party OSS source code and components within the target source code, various issues might arise like Compatibility[22], Interoperability[1], Testability [21], Performance [18], Privacy [6] and documentation [2]. All these issues are commonly faced by most of the designers. While integration, the designers might face various OSS issues like Compatibility, Interoperability, Testability and Documentation. The designers might also face some Performance and Privacy related issues after source code integration.

In order to overcome all above mentioned issues, an Integration model named InM (Integration Model) is proposed which enable all the designers to tackle different integration, incorporation issues in manageable and manner able way during designing phase.

The rest of this paper is organized as follows: Section II provides the related work done on the open source issues and models, coordination and collaboration among designers within the open source. Section III discusses various issues that might arise when integrating different OSS source codes or Components with target software. Our proposed model along with its detailed description is presented in Section IV. In Section V, an example case study has been devised in order to verify our proposed model. Section VI covers the conclusion and future extension of our research work.

2. Literature Review

Today, due to rapid growth in open source software development projects [4, 23], open source has turned to become one of the biggest software business. There are many circumstances where designers during their project or software development are incorporating different OSS source codes or components belonging to other designers.

There are several researches which has been made till now in which many researchers has provided different ways or methods for collaborating and coordinating in open source environment. Kumiyo Nakakoji et al. [15] have developed a framework for participative system. Their presented framework is capable of characterizing the evolution of an OSS community through changing the participants' perceived value and types of engagement. They have tested and reported their hypothesis on a case study named "GIMP Development Mailing List".

Authors in [7] seek to explore the conflicts that results in breaking down collaboration among designers in OSSD community. They explored the negotiation of conflicts like Leadership and control sharing across organizations and individuals in and between communities, collaborative efforts, and leadership and control structures in the Netbeans.org community.

Some authors have contributed their efforts towards the concept of virtual teams in open source development community. Katja Harej et al. [8] has introduced the preliminary proposal in their research for finding out if the virtual teams in environment of open source software development are using any of project management principles. Within the preparing phase of the research they have focused on open source virtual communities that work on larger projects for information system development. They have also formed research questions that will guide us in the future research. Their research work concludes with specific research needs in the study of virtual teams and their knowledge and use of project management principles.

Emad Shihab et al. [20] studied the IRC meetings among designers from two large open source projects along three dimensions: meeting participants, content, and style. Their study highlights the usefulness of studying IRC meetings. During their study, they found 1) that a small and stable number of the participants contribute the majority of messages in meetings, 2) that there are commonly discussed topics as well as project specific topics 3) that meeting styles vary across different projects.

Thomas Osterlie et al. [17] have proposed a model for maintainability in System Integration (SI) projects. System Integration projects are such kind of projects in which the maintainers have no direct access to the source code of the third-party software being integrated. In their model, they have focused on post-release activities instead of pre-released activities which are the part of traditional development. Their proposed model describes maintainability as a process characterized by ambiguity and negotiation that is supported through an infrastructure of debugging and coordination tools. They have also described the process of how the system failure can be managed in SI. On the other hand, Jean-Marc Lacombe et al. [11] have proposed an open source framework named FUDAA which is dedicated to the integration of scientific simulation code in homogeneous, graphic, and communicating interfaces. In FUDAA, simulation code exchanges are used as distributed software components that can communicate with each other through the network using a common data model.

Authors in [10] have addressed the impact estimation of 3rd party components of evolving nature integrated into long-living industrial software systems. They have presented their approach to use static code dependency analyses to identify explicit and hidden 3rd party component changes that might be propagated into the self-implemented software and requires further investigation. Further on, they have included code quality as well as bug tracker analysis for assessing the quality trend for multiple versions of the 3rd party component. They have identified 7 methods that are potentially impacted by changes of 3rd party components despite the absence of interface changes.

Apart from all these researches , some authors have tend to make various researches in which they have developed different frameworks for open source software development projects. Some researchers have proposed and defined the process of quality assurance within OSSD. Authors in [16] have proposed a Quality Assurance framework for OSSD models. Main aim of the research conducted by these authors was to investigate QA processes and to show their interactions in a process model. A qualitative research approach was selected and structured interviews with mature OSS projects were conducted to explore their QA practices. Their findings contribute to the development of a QA process framework. They used case study research in order to validate their research findings and show the correlation of QA key processes to project success.

Some authors have defined the process of Quality Assurance under OSSD by reviewing the literature of process of the latest quality assurance, under open source software development methods and techniques. The result of their reviews showed that how the process of quality assurance of open source software can affect the overall quality assurance principal [9].

On the other hand, some researchers have proposed different open source models and system. In [12], authors have developed a revision control system named CoxR that is capable of retrieving or crawling the development histories. This system creates software development information web which consists of designers, emails, and program deltas, and provides an interface to search, navigate, browse, and retrieve past development results. Authors have validated their system through a case study. The results of case study confirmed that CoxR helps designers to solve their problems by making it easier to search development history.

Some researchers have also studied open source standards [1] to find out the best ways of for enabling interoperability among between different technologies and applications. The role of open standards in interoperability has been analyzed and some of the policies introduced by the European Union for the use and dissemination inside Members States are examined. Additionally, the use of open source software combined with open standards has been presented and its major social benefits and economic impacts are highlighted. Apart of from this, some researchers have proposed a framework for peer to peer applications [21]. The proposed framework is based three-tier model of client-server paradigm and OSI model of distributed system paradigm. The goal of this framework is to fulfill and accommodate various requirements like Simplicity, Flexibility, Maintainability, Extendibility and Testability. Main focus of researchers in this framework is to provide easy and simple testing of new components in the real world.

In addition to all the above researches, some authors have provided ways of resolving compatibility issues within virtual environment [24]. Some researchers have proposed frameworks for improving performance in distributed multi-agent systems [18]. The proposed framework tends to focus on performance issues within distributed environment and provide a solution of how to improve the performance within this environment. Authors have also proposed some algorithms for reserving privacy in collaborative filtering systems [6]. The presented collaborative filtering algorithm is based on randomized perturbation techniques and secures multiparty computation. The randomized perturbation techniques are used in the course of user data collection and can generate recommendations with decent accuracy. Employing secure multiparty computation to protect the privacy of collaborative filtering by distributing the user profiles between multiple repositories and exchange only a subset of the profile data, which is useful for the recommendation. From the theoretical analysis done by the authors, it has been seen that the algorithm based on randomized perturbation techniques and secure multiparty computation not only protect the users' privacy, but also can keep the accuracy.

Authors in [2] have discussed some ways of how the open source model can be extended to the development of documentation. They have provided a framework for open-source documentation projects that illustrates what aspects of development need to be taken into account. Furthermore, open writing techniques and the current state of the profession in real open source documentation projects has also been examined.

From all the above mentioned researches, it will conclude that till now, no research have been made in Open Source Community where the researchers have addressed the integration issues that are faced by most of the designers when they integrate third party source code and components within their source code. So keeping in mind the previous studies in open source development, in this paper, an Open Source Integration Model has been proposed which enables the designers to tackle with the commonly occurred integration issues in a manageable and manner able way.

3. OSS Integration Issues

This section presents some common issues that are faced by most of the designers during and after integration of 3rd part components and code. As discussed earlier, the designers might face several issues when integrating third party source codes and components within their target source code. Some of the issues commonly faced by most of the designers are explained along with their description in Table 1.

Table 1. Integration issues faced by designer.

Integration Issues	Description
Compatibility[24]	The extent to which third party source code or component could work appropriately when merged with the target source code i.e. no platform, hardware or software issue.
Interoperability[1]	Measurement of the ability or ease with which third party source code or components interact with the target source code.
Testability[21]	The difficulty found when testing the target source code after integrating third party source code or components i.e. measuring the effort needed to verify the system change.
Documentation[2]	Ensuring that the documented procedures and guidelines are being followed while integration.
Performance [18]	Measurement of target source code's compilation time and execution speed after integrating third party source code and components with the target source code.
Privacy [6]	Ensuring that the final source code (i.e. after integration) performs as it is intended to do.

4. Proposed Integration Model-InM

In this section, a model named InM (Integration Model) has been proposed which addresses various issues that are faced by designers during and after integration of 3rd party components in their code. After studying various open source models [11, 10, 16], it is found that these models don't tackle with the problems that are faced by the designers during integration of third party source code. In this paper, a cyclic model has been proposed whose main purpose is to enable the designers to overcome all the integration issues that are discussed in Section III. This model lies in between two design phases of the design engineering process i.e. Component Design and Data Structure and Algorithm design. The model's integration process have two cycles one handling of issues during integration and the other capturing and managing integration issues after integration The proposed model comprises of various phases as shown in the Figure 1.These phases comprises of activities within which all the discussed issues are resolved.

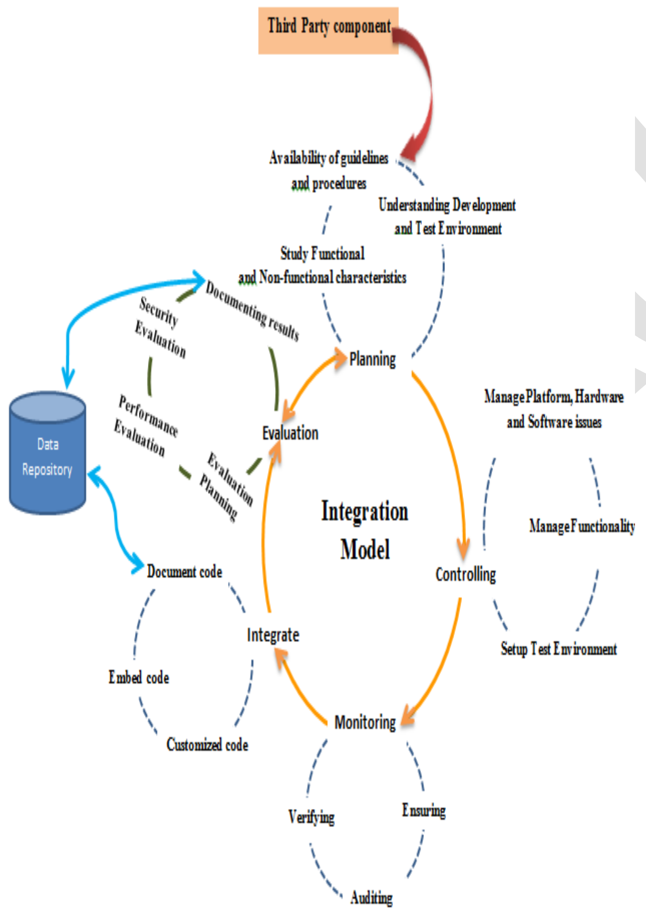

Figure 1. InM(Integration Model).

The above mentioned model is divided into two main sections 1st is 3rd party source code and component details and 2nd is Integration and evaluation process, constraints.

The basic purpose of proposing InM cyclic model is to facilitate the designers by providing them the clear path through which they can be able to handle different open source software development integration issues with different 3rd party components.

4.1. 3rd party source code and component details [10]

It is the initial activity and 1st section of the proposed model. In this section the designers embed third party source code and components from the repository according to their need. Before embedding it designers have to follow some integration phases in order to overcome the issues discussed in section III. There are three phases [3] which are:

- Planning ,
- Control and
- Monitoring.

4.1.1. Planning

Before embedding third party source code and components, the designers first need to plan how to integrate that source code and components within their target source code? To answer this question, the planning phase comprises of various activities. By following these activities, the designers can plan how to overcome the integration issues discussed in section III. In the first activity, the third party's procedures and guidelines must be made available to the active designers. This is because the designers must have understanding of those Procedures and Guidelines. In the second activity, the designers need to understand all the specifications related to development and test environment (i.e. Hardware, Software, Platform and Test Cases) provided by third party designers. According to our research, in OSS world, only 25-35% of designers provide ease to other designers by providing documentation for the system's source code and its components which they have developed [4], [25]. So it is very important that all the third party designers around the world must provide some necessary details like development platform, functionality, hardware and software used along with specification and brief description regarding test environment as well as test cases. By providing these details, the designers can easily plan to embed third party source code within their target source code. These two activities enable the designers to plan how they can resolve the Compatibility and testability issue that arises during source code integration.

In the third activity, the designers need to study the functional and non-functional characteristics of third party source code and components. To do this, the third party designers must provide a brief description regarding the functional and non-functional attributes of their particular piece of source code and components. According to our research, only 10 - 15 % designers facilitate other new designers by providing a brief description of functional and non-functional characteristics associated with their system's source code and components [22]. So, third party designers must provide such kind of knowledge to the designers so that they can easily plan to overcome the Interoperability issue that arises during source code integration. At the end of these activities, overall planning is evaluated during evaluation phase.

After following all these planning activities, it could be said that during this phase, active designer's learnability is also enhanced. The overall activities used for planning to overcome Compatibility [24], Interoperability [1] and Testability [21] issue are shown in Figure 2.

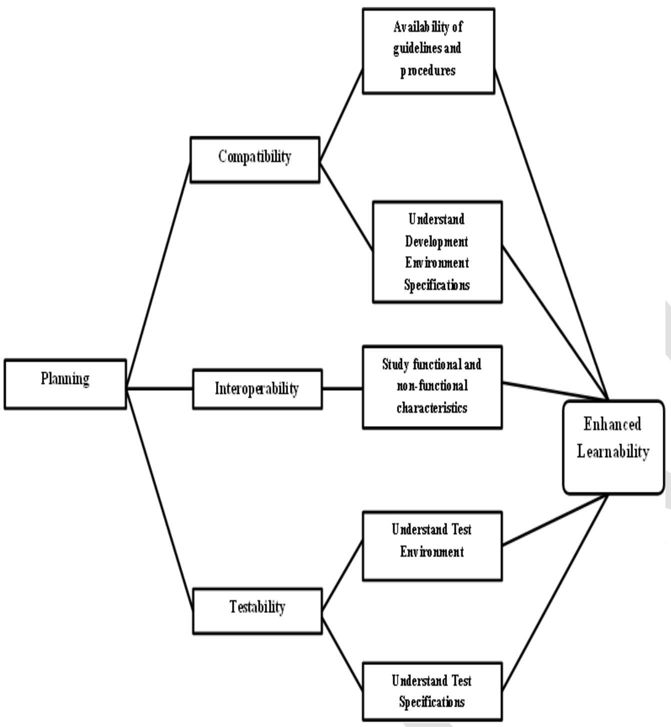
Availability of guidelines and procedures
Compatibility
Understand Development Environment Specifications
Planning
Interoperability
Study functional and non-functional characteristics
Enhanced Learnability
Understand Test Environment
Testability
Understand Test Specifications

Figure 2. Planning phase activities.

4.1.2. Control

After having complete planning regarding Integration issues, this phase describes how the designers can manage these planned issues? During this phase, the designers follow several activities associated with particular issue. The first activity of this phase enables the designers to manage compatibility issue within their source code. As during planning phase, the designers are provided with the understanding of the development environment, in this phase the designers manage platform, hardware and software issues according to their learning regarding development environment. There are some cases where designers can manage the compatibility issue without exerting extra effort and time. For example the active designer is working on .Net environment and he/she needs to integrate the third party component which is developed in J2EE platform. In this case, the active designer will not face any compatibility issue because J2EE framework is easily compatible with .Net environment. So, in order to manage integration issues, the designers must have clear understanding of third party's procedures, guidelines as well as development environment. With the deep knowledge of each environment, the designers can easily manage compatibility issues while integrating third party source code within their target source code.

The second activity of this phase enables the designers to effectively manage the third party source code and component's functionality within their target source code. After having complete knowledge of the functional and non-functional characteristics of third party source code and components, the designers can easily embed the function/methods being used by the third party designers within their source code and components and can easily manage those functions/methods within their target source code. Having clear understanding of third party designer's functionality, the designers can easily carry out effective communication between their and third party designer's source code and components. Therefore, it could be said that the designers can easily manage the interoperability issue arising during source code integration.

After having the understanding of test environment and test specification during planning phase, during third activity of this phase, the designers setup their own test environment in order to test their target source code when merged with third party source code and components. Active designer establishes his/her environment keeping in mind the third party designer's environment and test specification. With the clear understanding of third party designer's test environment and specifications, the designers can efficiently manage the test cases developed by third party designers within their target source code. Hence, it could be said that the designers can easily manage the testability issue arising during source code integration depending upon their knowledge regarding third party designer's test environment.

After following all the activities of this phase, it could be said that active designer's management capabilities are enhanced. This betterment is due to the management of third parties development and test environment and management of their source code and component's functionality within their target source code. The overall activities used for controlling Compatibility [24], Interoperability [1] and Testability [21] issue are shown in Figure 3.

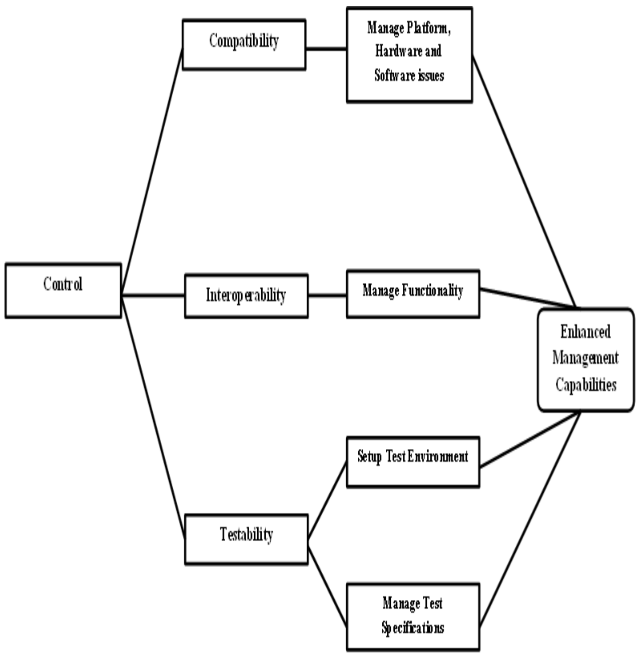
Control
Compatibility
Manage Platform, Hardware and Software issues
Interoperability
Manage Functionality
Testability
Setup Test Environment
Manage Test Specifications
Enhanced Management Capabilities

Figure 3. Control phase activities.

4.1.3. Monitoring

After managing all the issues in control phase, the monitoring phase[3] describes how the designers can continuously monitor that their source code when merged with third party source code and components is no more victimized by the compatibility, interoperability and testability issues? This phase comprises of several activities. In the first and second activities, the designers checks and guarantee their source code after merging third party source code and components by executing it twice a week. After checking, the designers can easily ensure that the compatibility and interoperability issues has been resolved and the final target source code works as it is intended to do. Once these two issues are resolved, it could be said that in the third activity, the designers can easily resolve testability issue by ensuring that no environment conflict exists and can easily verify that the test cases developed for target source code (when merged) generates appropriate output as they are intended to do so.

After following all the activities of this phase, it could be said that active designer's experience is enhanced. This is because when the designers monitor their source code twice a week, every time they perform this monitoring task in very less time as compared with one formed first time. The designers get experience by using several tools (Fit [26], BizUnit [27], Nester [28], NUnit [29], benerator [30] and many more [31, 32, 33]) during testing their source codes. The overall activities used for monitoring final target source code in order to overcome Compatibility [24], Interoperability [1] and Testability [21] issue are shown in Figure 4:

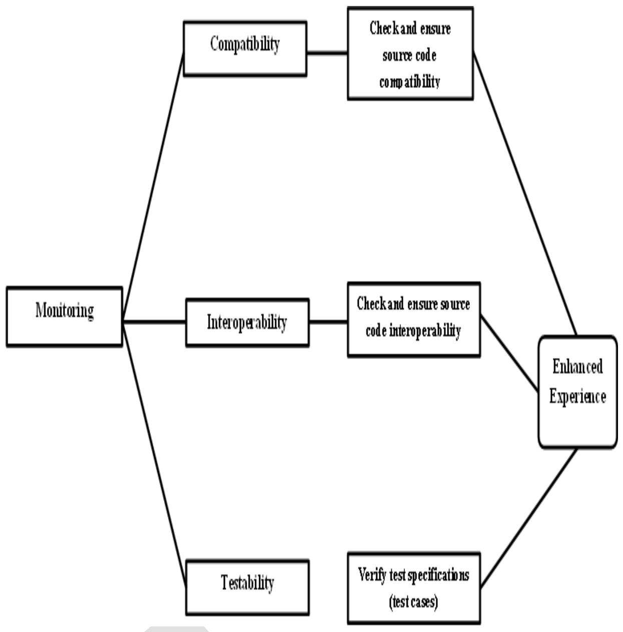
Monitoring
Compatibility
Check and ensure source code compatibility
Interoperability
Check and ensure source code interoperability
Testability
Verify test specifications (test cases)
Enhanced Experience

Figure 4. Monitoring phase activities.

4.2 Integration and Evaluation Constraints

After completing all the phases and their associated activities and getting solution to all the issues, in the 2nd section of this model third party source code and components is embedded and documented with customized source code [13]. The Integrated source code is then stored in the repository. When this integrated code is compiled and executed, some issues might be arising like reduced performance [18] and secrecy leaks (privacy [6]) issues. In order to overcome these issues, the designers need timely evaluation (weekly) of the integrated target source code and record the monitoring results in order to ensure source code's performance by measuring the compilation time as well as processor speed. To overcome secrecy issue, the active designer using the same record can ensure that the source code performs its intended functionality every time when the source is compiled or executed. The performance monitoring results generated as a result of monitoring an integrated source code on weekly basis is illustrated in the form of statistics as shown in Figure 5 of next section. Once these issues are resolved, these results are documented.

5. Model Verification and Discussion

In this section, a case study entitled as Moodle [14], [37] has been picked which is an open source Web application (released under the GNU General Public License) designed for producing Internet-based courses and websites. It is a Course Management System (CMS), also known as a Learning Management System (LMS) or a Virtual Learning Environment (VLE). It is a free web application that educators can use to create effective online learning sites. This application helps to people to create dynamic sites where learning communities can communicate and collaborate. It is written in PHP and runs without modification on Unix, Linux, Windows and any other systems that support PHP and a database, including most webhost providers. This case study has been taken in order to verify the above mentioned integration model. This case study comprises of multiple designers with several different modules.

- Assignment submission
- Discussion forum
- Files download
- Grading
- Moodle instant messages
- Online calendar
- Online news and announcement (College and course level)
- Online quiz

Consider the scenario where the designers want to customize some components along with the existing modules. Out of many components these three components are selected for discussing the integration problems.

- Videowhisper [34] supported by Joomla and is used for video conferencing
- DotNetOpenMail [35] supported by .Net framework and is used for emailing
- GXSMS [36] works with all versions of Visual Basic and VB.NET and is used for sending by using a modem and receiving SMS

The integration problem arises when the designers web site want to add GXSMS and DotNetOpenMail component code for their frame work. It was found that these components are incompatible with .net and it is very time consuming to customize it into .net with in the current time frame or to select any alternative at this level. The same problem arises for Videowhisper when it is customized and integrated in the actual code. For tackling these problems the designers faces project delays and late submissions. The quality of the project was also getting compromised at different levels because of meeting deadlines hurriedly. For overcoming such problems the designers team uses the above mentioned integration model during the development of their new project which is a social networking web site. The same components on the same development platform were used. The results are tremendously successful.

Because of the proposed frame work the designers learn, understand about the alternative component and getting all the necessary details which are supposed to be understood by the designers during planning phase activities as shown in Figure 2. Now once the designers have successfully planned how to overcome the arisen integration issues during planning phase, in the next phase, they cope with those planned issues by managing platform, hardware, software, functionality and test environment. In planning phase, the designers have complete understanding regarding the development environment, function/methods used and test environment of third party source code and component, the designers can now easily embed this source code within their source code and can effectively manage the integration issues in a manageable and manner able way. Once these issues have been managed, in the next phase, the designers can easily check, ensure and verify that these issues no more exist within their source code when integrated with the required component.

For checking the performance and secrecy issues of the code the designers execute the integrated source code on weekly basis and recorded the results in the form of statistics as shown in figure 5. According to the statistics, the compilation time of integrated source code is improved in every week and its percentage lies between the 60-90%. On the other hand, the speed of the processor on which the source code is executed is also improved every week and its percentage lies between 50- 80%.

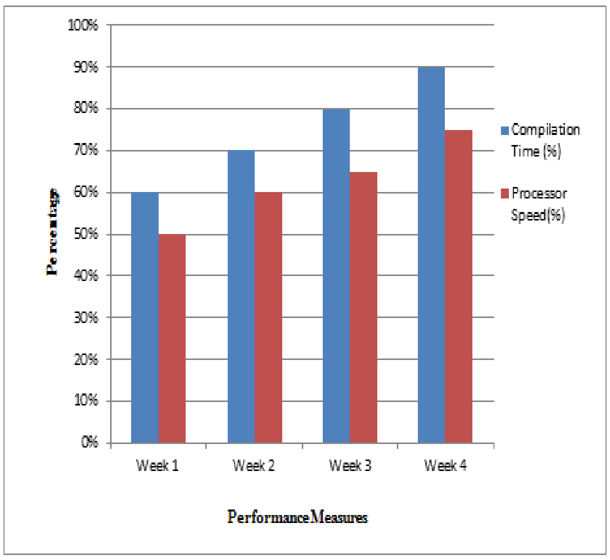

Figure 5. Weekly basis performance measures.

6. Conclusion and Future Direction

Open source software projects have gained worldwide acceptance in the recent era. Open source software development has now a day became biggest software business. It has facilitated every designer within or outside the open source community by providing free access to every kind of OS projects, their source code and even their components. Today, most of the designers who are working on the development of open source projects need to integrate third party source code and components within their projects. The designers might face several issues during and after source code integration. These issues arise due to the lack of sufficient details regarding third party's development and test environment. So in order to overcome the issues, in this paper an integration model named InM has been proposed which addresses various issues that the designers might face during and after source code integration. During the integration of third party source code and components the designers faces various like Compatibility, Interoperability, Testability and Documentation. After integrating third party source code and components with the target source code, the designers faces some issues related to performance and privacy. InM model facilitates all the designers to integrate third party source code and components in the manageable and manner able way.

In the future, this model can be extended for integrating source code of agile open source projects where the other stakeholder's (direct or indirect) feedback plays very important role throughout the development life cycle.

References

[1] Almeida F., Oliveira J., Cruz J., "OPEN STANDARDS AND OPEN SOURCE: ENABLING INTEROPERABILITY" *in International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.1*, January 2011

[2] Berglund E., Priestley M., "Open-Source Documentation: In Search of User-Driven, Just-in-Time Writing" in *SIGDOC'01* ACM, 2001

[3] Chang S., Lee J., Yi W., "A Practical Management Framework for Commercial Software Development with Open Sources", *IEEE International Conference on E-Business Engineering*, 2010

[4] Deshpande A., Riehle D., "The Total Growth of Open Source" *in Proceedings of the Fourth Conference on Open Source Systems, Springer Verlag*, 2008

[5] FellerJ., FitzgeraldB., "A framework analysis of the open source software development paradigm" *in Proceedings of ICIS,* 2000

[6] Gong S., "Privacy-preserving Collaborative Filtering based on Randomized Perturbation Techniques and Secure Multiparty Computation" *in International Journal of Advancements in Computing Technology, Volume 3, Number 4*, May 2011

[7] Jensen C., Scacchi W., "Collaboration, Leadership, Control, and Conflict Negotiation and the *Netbeans.org* Open Source Software Development Community*" in Proceedings of the 38th Hawaii International Conference on System Sciences*, 2005

[8] Harej K., Horvat R.V., "Project Management Principles and Virtual Teams for Information Systems Development: Preliminary Proposal" *in Proceedings of the ITI 2007 29th Int. Conf. on Information Technology Interfaces,* June 25-28, 2007

[9] Khanjani A., Sulaiman R., "The Process of Quality Assurance under Open Source Software Development" *in IEEE Symposium on Computers & Informatics*, 2011

[10] Klatt B., Durdik Z., et.al., "Identify Impacts of Evolving Third Party Components on Long-Living Software Systems" *in 16th European Conference on Software Maintenance and Reengineering*, 2012.

[11] Lacombe J-M., Pasteur O., "FUDAA: An open-source framework for the integration of simulation codes, pre-processing, and post-processing tools" *in IEEE International Conference on Signal Image Technology and Internet Based Systems*, 2008

[12] Matsushita M., Sasaki K. , Inoue K., "CoxR: Open Source Development History Search System" *in Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05)*, 2005.

[13] Merilinna J., Matinlassi M., "State of the art and practice of open source component integration", *Proceedings of the 32nd Euromicro Conference on Software Engineering and Advanced Applications, IEEE Computer Society*, pp. 170–177, 2006.

[14] Mihailescu E., "An Overview of Open Projects in Contemporary E-Learning: A Moodle Case Study" *in Studies in Computational Intelligence*, Volume 217/2009, 2009

[15] Nakakoji K., Yamada K., Giaccardi E. "Understanding the Nature of Collaboration in Open-Source Software Development" in Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC'05) 2005

[16] Otte T., Moreton R., Knoell H.D. "Development of a Quality Assurance Framework for the Open Source Development Model" *in The Third International Conference on Software Engineering Advances,* 2008.

[17] Osterlie T., Wang A.I., "Establishing Maintainability in Systems Integration: Ambiguity, Negotiations, and Infrastructure" *in 22nd IEEE International Conference on Software Maintenance (ICSM'06),* 2006

[18] Pandey T.N., Panda N., Sahu P.K., "Improving performance of distributed data mining (DDM) with multi-agent system" *in IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 3*, March 2012

[19] RaymondE.S., "Linux and open-source success". *IEEE Software*, 16(1), pp. 85–89, 1999.

[20] Shihab E., Jiang Z.M., Hassan A.E.," Studying the Use of Developer IRC Meetings in Open Source Projects" *in Proc. ICSM*, 2009
[21] Shamsaie A., Habibi J., Ghassemi F., "TIERPEER: A THREE-TIER FRAMEWORK FOR P2P APPLICATIONS" *in IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.2*, February 2007
[22] Von G., et.al., "Special issue on open source software development" in Research Policy 32 (2003)
[23] Walli S., Gynn D., Rotz B.V., "The Growth of Open Source Software in Organizations" *under a Creative Commons Attribution 2.5 License*, 2005
[24] Wright T.E. ,Madey G., "A Survey of Technologies for Building Collaborative Virtual Environments" *in The International Journal of Virtual Reality,* 2009
[25] Open source Initiatives, [Online] Available: http://www.opensource.org/docs/osd. [14th June 2012]
[26] http://fit.c2.com/ [27th June 27, 2012]
[27] http://bizunit.codeplex.com/ [27th June 27, 2012]
[28] http://nester.sourceforge.net/ [27th June 27, 2012]
[29] http://www.nunit.org/ [27th June 27, 2012]
[30] http://databene.org/databene-benerator [27th June 27, 2012]
[31] http://java-source.net/open-source/testing-tools [27th June 27, 2012]
[32] http://csharp-source.net/open-source/testing-tools [27th June 27, 2012]
[33] http://www.opensourcetesting.org/unit_dotnet.php [27th June 27, 2012]
[34] http://www.videowhisper.com/?p=Video+Conference[28th June 27, 2012]
[35] http://dotnetopenmail.sourceforge.net/ [28th June 27, 2012]
[36] http://www.gurux.fi/index.php?q=GXSMS [28th June 27, 2012]
[37] http://moodle.org [18th July 18, 2012]

Mariam Nosheen is working in the Department of Computer Science, Lahore College for Women University, Lahore, Pakistan. She is supervising various development projects and research thesis at under-graduate and graduate levels. Her research interests are in the domain of Human Computer Interaction, Software Engineering and Mobile Computing.

Zahwa Syed received her Bachelor's degree from Lahore College for Women University. She is currently doing Masters in Computer Science with specialization in Software Engineering from the same institution. Her research interests are in the domain of Software Engineering, Networking and Mobile Computing.