ABSTRACT

It has been noticed that the open-source software (OSS) systems is well contributed and used in the developing countries. Though this technique is adopted globally in a successful manner and in all sectors on businesses. Open-source software (OSS) creation is accepted by banks in large scale and the importance is well understood. The complete study has been divided into two different parts covering different aspects in relation to each other. First Part was the critical assessment of Open Source Software and other was designing a new Open Source Software. Tools used to develop hybrid open source software were VISUAL STUDIO 2008, SQL SERVER 05 and Microsoft Chart Controls. In the critical analysis observed properties and the points to be required for the well developed methodologies for the success of OSS in the developing countries were discussed. A new software was designed as the requirements of the now running loan systems in the banking system looking all aspects of benefit of the customer, employee and the manager.

Key Words: Open Source Software, Design ,Banking System and Loan Applications

INTRODUCTION

Banks also seems to be the best way of managing the financial deals of the common pupil than the private financers. The best importance of banks is existed in favoring the loan deals to the required individuals. This study is searching out the best implementation of Computer engineering in providing the easy access to financial assistance through open source software for the benefit of the customers and banks. It has been noticed that the open-source software (OSS) systems is well contributed and used in the developing countries. Though this technique is adopted globally in a successful manner and in all sectors on businesses. Open-source software (OSS) creation is accepted by banks in large scale and the importance is well understood. open-source software (OSS) actually go the logic of closed-source software (CSS). Closed-source software (CSS) is produced in a system of collaboration, instead of concentrating research and development inside one firm. Individual programmers around the world contribute to the code of the program, which is freely available on the Internet. In the existing technological accessibility these all can be named as a de facto global activity. It can be said that any individual with just the minor skills of programming can access this approach upto any contributory level from any part of the world.
the study involves the specific issues to in relation to new designed and selected existing open source software.

Methodology of Designing Techniques

The Open source software that developed with respect to the banking system in relation to loan section only. An open source software means it should be platform independent. By saying platform independence we mean that on whatever platform it has been created, it should support all the platforms. Also there should be ease of modifications in the software. A lot of open source software are available like the Linux operating system. Its version gets changed in every three month. The reason behind the version changed is it is open source. Developers from all over the world develop applications for this platform. Open source software is computer software that has a source code available to the general public for use as is or with modifications. This software typically does not require a license fee. There are open source software applications for a variety of different uses such as office automation, web design, content management, operating systems, and communications. The key fact that makes open source software (OSS) different from proprietary software is its license. As copyright material, software is almost always licensed. The license indicates how the software may be used. OSS is unique in that it is always released under a license that has been certified to meet the criteria of the Open Source Definition.

Tools Employed

The following tools were used to develop hybrid open source software.

I) VISUAL STUDIO 2008
II) SQL SERVER 05
III) Microsoft Chart Controls

IDE Features Correspond to Project Capabilities-
Features in the IDE, such as Add Reference and the Toolbox, now correspond to the version of the .NET Framework or Silverlight that your project targets. As a result, types, members, and controls from assemblies that target an earlier version of the .NET Framework do not appear in IntelliSense and produce background compilation errors. For example, if your project targets .NET Framework 2.0 and you use a language feature that .NET Framework 2.0 does not support, such as anonymous types, the IDE will mark the code as an error.

Test-First Support-
In Visual Basic or Visual C#, the IDE can now generate code stubs for new types and members from their usage before they have been defined. As a result, you can write tests first, and then generate the code that is required to compile the tests. Additionally, IntelliSense now provides a suggestion mode that prevents IntelliSense from automatically completing a type or member that has not yet been defined.

Application Lifecycle Management-
The Application Lifecycle Management client application is now available in all versions of Visual Studio (excluding Express versions). Application Lifecycle Management includes version control, work item tracking, build automation, a team portal, reporting and business intelligence, Agile Planning Workbooks, and test case management—everything that you need to collaborate on your software development projects. Application Lifecycle Management is included in Visual Studio 2010 with one Client Access License (CAL). You can purchase additional Client Access Licenses as needed.

Code Generation and Text Templates-
Text templates can be created and used in any version of Visual Studio, without the need for any other component. Generating any kind of text file from your application has been made easier in Visual Studio 2010 with the introduction of Preprocessed Text Templates. Support for code generation has also been improved through better integration with the build system, so that your generated source code is always up to date after any change to the source model.

Start Page-
The Visual Studio 2010 Start Page features a new look and new functionality. A tabbed content area links to a variety of selected and categorized learning resources. This includes MSDN resources, community resources, and a customizable news feed. An improved Recent Projects list lets you add projects to the list or remove them with a single mouse click. You can also install custom Start Pages from Extension Manager.

Extension Manager-
You can find and install Visual Studio extensions from the IDE by clicking Extension Manager on the Tools menu. Extension Manager downloads and installs community-published extensions from the Visual Studio Gallery Web site without requiring that you open a browser. It also enables you to uninstall, disable, or re-enable installed extensions.

New Project Dialog Box-
The New Project dialog box in Visual Studio 2010 includes the same search and installs capabilities as the Extension Manager, except that it searches only for project templates and item templates.

New Help Viewer
The Help Viewer has been completely redesigned for Visual Studio 2010. You can view documentation online or offline by using your preferred Web browser, download the latest documentation on demand, navigate the documentation by using a simplified table of contents, search for specific content by using an improved full-text search algorithm, and use a more streamlined F1 system to locate content specific to the task that you are working on.

**Visual Basic and C# Languages**

In Visual Studio 2010, the Visual Basic and C# languages continue to move toward feature parity. This enables you to choose a language based on personal preferences because both languages are equally capable. This section lists some of the new features in C# and Visual Basic.

**Visual Basic Language**

Visual Basic has new features that shorten syntax and enable you to write code faster. These features include auto-implemented properties, implicit line continuation, collection initializers, and single- and multi-statement lambda expressions. Additionally, Visual Basic now supports simplified deployment through type equivalence.

**Results:**

Creating open source software for any banking system involves a lot of attention related to the banking environment. Open source software is software which is always open for the modification. As the objectives were and design a new software. Critical anlysation proved as a mile stone in getting new dimensions and understanding the open source software.

Observed analysis of Open Source Software involves the Properties. Investigators personal observations are mentioned below. Open Source Software (OSS) is software that is legally free to distribute and modify. The ability to modify the software is an important distinction from what is known as freeware. There are some main arguments as to why OSS is better than alternatives:

1. Low Development Cost
2. source code is made freely available
3. Easy Access
4. Open Modifications
5. Demand in Competitive Markets
6. Socially Acceptable
7. Better Software understanding

Fundamentally, OSS is inexpensive because the code is cost free. There is some argument about this because one may have to pay for bandwidth usage or a CD, etc., but the actual code does not have a price tag. However, “free as in price” is not the only way in which OSS is less expensive. A competitive market can drive a better product (as Firefox has forced improvements in Internet Explorer); it can also drive the price down (Wheeler, 2007). Lastly, OSS is driven on volunteer labor. This is not to say big projects like Linux and Solaris do not have paid staff working on the code, but being able to leverage any amount of volunteer labor keeps costs down (Byfield, 2005). The OSS model has allowed Red Hat to be #1 in enterprise software according to CIO Insight the last four years and #1 overall three of the past four years (Red Hat, 2008).

1. Easy and Free Further Access to License:
   In the Open Source Software the license shall is not restricted from selling. This approach is not restricted to giving away the software as a component of an aggregate software distribution containing programs from several different sources. Open Source Software The license shall not require any royalty or other any kind of complementary or compulsory fee when sailed.

2. Presence of Source Code:
   Open Source Software approach always involve program with a source code and the technique always allows approach to respective source codes of the program. In open source methods distribution in source code is commonly presented in compiled form and some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost–preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately highlighted source code is not ethical. Intermediate forms such as the output of a Pre-processor or translators are not allowed.

3. Modify Approach As Required:
   Open Source Software approach always involves modifications of the derived works as and when required.

4. Distribution of Modifications:
   Open Source Software approach very positively allows the distributed under the same terms as the license of the original software of the modifications done.

5. Authentication of further Approaches or Distribution:
   The developed open source software license clearly allows the permission of further distribution modified source code. This process actually requires derived
works to carry a different name or version number from the original software.

6. Dignity of Work Done:
   In open source software methodologies distribution of 'patch files' with the source code is essentially required in order to openly perform any modification in the existing program at the build time.

7. Favorable Approach:
   Open source software license never favor specifically any individual or a community. It always exists in a open access approach.

8. Decentralization:
   Open source software license always work in a decentralized atmosphere and must not restrict anyone from making use of the program in a specific field of endeavor.

9. No Discrimination:
   Open source software license specifies no discrimination in the use and application of software whether used for research, business or any other field.

10. No Specified Distribution of Rights:
    In the process of distribution and redistribution all the rights attached to the program must apply to all without the need for execution of an additional license.

11. No specialized production of license.
    The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

12. The License Must Not Restrict Other Software
    The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

13. The License must be technology-Neutral Approach:
    The respective License is always technologically neutral. It means that no provision of the license may be predicated on any individual technology or style of interface. (The Rationale section in the original text is deleted).

14. No Zone Specification:

Open source software application must be always no zone specified as and when required the modification is done to make it applicable in any technological zone or foundational background. It is well adoptable in all desired applications.

**Designed Open Source Software**

After finalizing the procedure or methodology of designing the hybrid Open Source Software through the use of Vb.Net and My sql. The designed was done to achieve the objective of Designing a new software with respect to the banking system in relation to Loan section. Below mentioned the explanation along with the original snap shots of the developed open source software

Figure 1 shows the initial login of the Developed Open Source Software. In this form two options ‘Login System’ and ‘Comparisons’ appeared with further actions. When approached to ‘login System’ three further options appear, classified as ‘Manager’ ‘Employee’ and ‘Customer’ mentioned in figure 1(a) with further applications.

Fig. -1: Snap shot of Login Form.

![Fig. -1: Snap shot of Login Form.](image1)

Fig.-1(a) Snap shot of extensions of login system in Login Form
Fig. 2 Indicates the existing user Id and Password of the respective login user, (In this case login user is a Customer). If the login is correct the same user will only then be welcomed and allowed for further proceeding shown in figure 2 (a). When approached to ‘My Account’ four further options appear, classified as ‘Check Previous Loan Status’ ‘Apply for new Loan’ ‘Change My Password’ and ‘Loan processing Verification’ mentioned in figure 2(b) with further applications.

Fig.-2 Snap shot Customer Login

- Fig.-2(a) Entry to the Account after correct login by the respective user.

Fig.-2(b) Access to the Account after correct login.

Fig. 3 indicates the after form of access from ‘my account’. Form in loan department requires to be filled in four different sections i.e. ‘Basic Information’ ‘Particulars’ ‘Loan Details’ along with the highlighted instructions that will guide the customer. This lead to the figure 3(a) showing application number for any further query.
Fig. 3 Snap shot of the loan application form.

Fig.-3(a) Snap shot showing application number after filling successful loan application form.

Fig. 4 Indicates the existing user Id and Password of the respective login user, (In this case login user is a Employee). If the login is correct the user employee will only then be allowed for further information and proceeding, shown in figure 4 (a). When approached further related information about the customer and processing applications like documents verification and access to manager appears. Further the information can be modified or updated (fig. 4(b)) and send to managers access file with all the required feedback (fig. 4(c)).

Fig.-4 Snap shot of Employee Login.

Fig.-4(a) Snap shot of customer information form.
Fig.-4(b) Snap shot of confirming updated done by the employee.

Fig.-4(c) Snap shot of confirming information sent to manager by the employee.

Fig. 5 Indicates the existing user Id and Password of the respective login user, (In this case login user is a Manager). If the login is correct the manager will only then be allowed for further information and proceeding.

Fig.-5 Snap shot of Manager Login.
Figure 5 (a) indicates the further approach of the manager after successful login. In this form manager receives feedback by the employee and all the related information about the customer along with the official customer application number and the amount required, for further necessary action. Manager can check every information in respect to the interest of the bank and reply the agent about the decision of accepting/rejecting or if any other information required in relation with the customers document verification. Further, the information can be updated (fig. 5(b)) and revert to the related agent for information and required actions. In this process manager receives the confirmation report.

Fig.-5(a) Snap shot of customer information and Employee feedback form sent to the manager.

Fig.-5b) Snap shot of updated/confirming information sent to agent by the manager.

Conclusions

By the Critical analysis it was concluded that OSS is indeed the start of a fundamental change in the software infrastructure Market place, and is not a hype bubble that will burst. Within five years, 50% of the volume of the software infrastructure market could be taken by OSS. OSS's position in large servers (e.g. those managing massive multi-user databases), such as those that underpin many large Government procurements, will grow from its current position of near zero
penetration, to a position where OSS is a viable option, within 2 - 3 years. Designed OSS was best applicable in the banking loan system and shall be the better option for all the bank users along with the comfort for the customers for fair and easy access. As we all know that software development and its implementation along with the testing is a field which will never end up. Case of modification would be always there. Our implementation is basically working on a concept called web services which works fantastic with the same communication protocol but we all know that this is world of advancement and in the upcoming few years, it would be mandatory to cope up with multiple protocol communication model system. That would require a deep understanding of the communication foundation and the service contract. With the implementation of the multiple protocol system communication in the open source software system, the execution would not just be fast but it would be kind of lighting fast and the performance evaluation would add extraordinary parameters which would really give a new hike to the open source software development environment.

References


