ISSN 2278-3091

Volume 10, No.2, March - April 2021

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1251022021.pdf

https://doi.org/10.30534/ijatcse/2021/1251022021

Model Architecture for Subversion Software Repositories System

Ammar Oad[1],Mir Sajjad Hussain Talpur[2], Yasmeen Noonari[2], Fauzia Talpur[3], Abida Luhrani[2], Akhtar Hussain Soomro[2], RaheelSarwar[2], Shakir Hussain Talpur[2], Gaofeng Luo[1]

[1]Faculty of Information Engineering, Shaoyang University, Shaoyang 422000, China, ammar_2k309@yahoo.com

[2]Information Technology Centre, Sindh Agriculture University, TandoJam, Pakistan,
mirsajjadhussain@sau.edu.pk

[3]USindh Laar Campus, Badin, Sindh, Pakistan, fozia.g.talpur@gmail.com

ABSTRACT

In the development cycle of any software, software evolution plays an important role. Software Evolution is essential and challenges characteristics in the field of software engineering. Many research studies have concluded that 60-80% of efforts are spent on the evolution and maintenance of the software. Software repositories, for example, version control and bug tracking systems, are vital fragments of numerous software maintenance activities. The version control system is an incremental version tracking system, introduced to avoid unnecessary overwriting of files such as programming code, web pages, and records. It also helps to decrease the confusion affected by duplicate or outdated data. In this proposed research, the SVN repository was maintained and analyzed for msitone.wikispaces.com to minimize the efforts as well as resources for future users. Two semesters' data was taken as the observations for the analysis. The obtained archives were implemented and deployed for postulating the local archives to the online server via the Visual SVN server, allowing remote users to access a repository.The tortoise SVN tool is used as an SVN client to access the repository of Wiki spaces. Analyses were observed with and without implementing the SVN repository. The result shows that implementing the SVN repositories is helpful for the maintenance of the Wiki spaces. It also reduces the cost, time and efforts for their evolution. Whereas without implementing the SVN repositories, Wikis paces were just supposed to be building the house by putting each brick from the start.

Key words: Model Architecture, Subversion Software, Repositories System

1. INTRODUCTION

Source control (version control) is a central component of the modern software development process. Various kinds of version control systems are available like CVS (Concurrent Version System), SVN (Subversion), Git, Mercurial, Bazaar, Libre-Source, and Monotone. SVN stands for subversion; it is a version control system, introduced to avoid unnecessary overwriting of files and decreases the confusion affected by duplicate or outdated data. CollabNet Inc developed a revision control system in 2000[1] it was used to maintain historical versions files as well as current files like source code, Webpages, and documentation. This open-source revision control system used to perform changes in text-based files i.e. source code. Subversion was designed to replace older CVN. SVN performs work on directories, files helpful to manage as well change them. This feature of SVN provides the facility to return over the prior version of your program at any point which was built at any time.SVN development is faster and less prone to errors. If any user enters the wrong code unintentionally and missed all the chances to undo the code, the versioning control system can resume the code from any point they want. Some of the basic command-line usages of SVN (subversion) are listed here;

1. Check
2. Create
3. Update
4. Commit

The repository has three types of sessions such as the trunk, branches, and tags. Trunks contain the latest development code, Branchespossess multiple versions of the same product lines for the development can be used to produce and Tags are any version is released to the public, with source code version number tag.

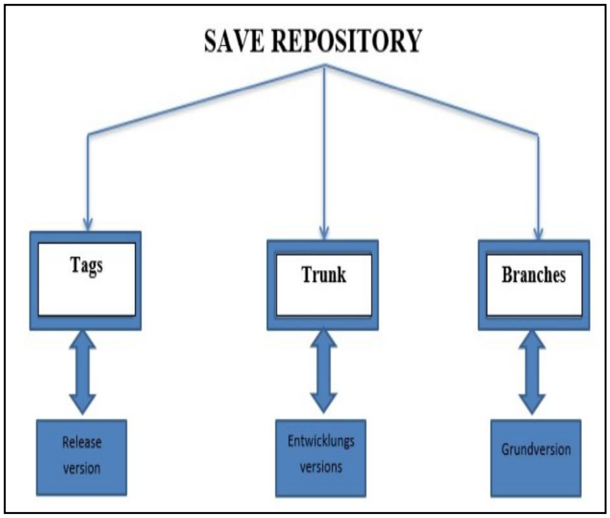


Figure1.1: General Concept of SVN repository

F/OSS (Free and open-source software) schemes participants use some tools like CVS and SVN version control systems for lists of people and bug tracking systems and so on. These types of tools serve as a repository, from which data can be extracted and how people participate in code commits and email posting.F/OSS (Free and open-source software) missions mostly use repositories well to identify the developer's involvement [2].

A repository is a piece of information about the database that is shared with engineered artifacts created and used by the enterprise. Examples of such a sample include software, documents and maps, information systems, and developed discrete components used by systems such as industrial plants, electronic circuits, automobiles, and airplanes. Storage services and tools are provided by a repository that helps developers they do not need to develop a tool-based database of the objects. A common repository allows instruments to share information, not including a common repository, and you will need a protocol exchange of information between machines[3],[4],[5].

The MSR (Mining Software Repository) launch is a significant challenge for both investigators and experts in the field of software repository mining, encourages them to meet its approaches, tools launch to (general) a common floor for assessment of PostgreSQL and AgroUML repositories proposed by datasets [6].

The subversion control system allows multiple people to work on a project at the same time and, for this purpose, it maintains the central repository, hosted on a server. That central repository contains the version of files and allows people to create local copies of their personal computers in a process called checkout. The repository is a central location where different types of data are stored and maintained. In other words, we can say that a repository is a position where various databases or files are positioned over a network. Visual SVN Server is used as a central and local repository that can be accessible to remote users. Visual SVN Server provides a facility to manage and install a multi-functional subversion server easily on the windows platform. Tortoise SVN is used as a client tool for maintaining the SVN repository by applying their multiple functions and commands. It manages the projects that are located in a subversion repository.

2. REVIEW OF LITRACTURE

The study promotes and highlights the international position of (OAR) open access repositories [7]. This research was wholly based upon those types of data that were collected from an open directory of free access repositories (DOAR). They collected data thoroughly investigated based on selected factors. Their study focused on stage of 2168 repositories only that were listed in the Open Directory of Open Access Repositories on February 8-10, 2012. Open access repositories are still in the early stages of development in well-developed countries such as Japan, Malaysia, India, China, and Taiwan; all of these countries contribute a significant number of repositories because they have become aware, but Japan has more than 100 contributing repositories.

[8] the first repositories were proposed in the early 1990s and have already emerged in a few areas, providing a valuable route for disseminating research results.Using very strict addition criteria, were 56 subject repositories recognized from a bigger number of indexes which contained two indexes of the repository, A The study showed a high variety of organizational size models, functions and topics. They first began to emerge subject repositories catered to the long-term market demand, but later developed an Internet search engine.

[9]Conducted Group at the University of Mannheim describes a large, unabridged data-set of Java source code collected and shared as part of the Merobase Component. They described the techniques to gather format and store the dataset, as well as the core abilities of the Merobase search engine like classic keyword-based, interface-based and test-driven search. This dataset represents one of the largest searchable collections of source and binary modules and these are available online and recently made available for download and use in additional research projects [10]. The making of searchable software for increasing reuse levels in software development heaving the web for reusable source files had restrictions since the found files are relatively isolated. Most open-source hosts excluded crawlers from their browsable repositories on the web (via robots.txt) so that using the locations of CVS and SVN repositories was a more sustainable solution [11]. Merobase's code repository was accessible to other researchers. They have created a downloadable zipped tarball file (compressed approximately 15 GB, unpacked nearly 50 GB) archived. They made the data set and index underlying the Merobase software search engine public in the hopes that researchers will find the artifacts contained useful and will develop novel, innovative applications for it [12].

[13]Studied how global the availability of quality student work researchers, practitioners, and educators can be attached to. The material properties and CAPSTONE papers download rate search machine optimization study the effects of factors. An online digital repository for public use, which is offered through the Texas State University MPA CAPSTONE all 290 papers reviewed [14]. Open access digital institutional repository universities engaged in an exchange of scholars using is a relatively new technology. Students find a research topic and their goal is to build the Texas State University the CAPSTONE process distinguishes itself from the two-course sequences.Students follow five research objectives: description, gauging the decision-making, or using one of the explanations should be. Results download rate search engine that shows strong support for the effects of factors [15],[16].

Conducted research that provided institutional repository managers with an overview of awareness and advocacy techniques because the repository is more useful in terms of regular updates of any academic record in Australia and internationally [17].He was finally receiving research results indicating that the repository provides more warmth for their daily workflow by providing appropriate assistance to education will be accepted if useful.

3. RESEARCH METHODOLOGY

3.1 SVN DESIGN FOR THE WIKI

Subversion is an open source (versions control system) that is used to manage old versions of data such as coding, website pages, and records.In the proposed research, the SVN repository was maintained and analyzed for msitone.wikispaces.com, to minimize the efforts as well as resources for future users. Msitone.wikispaces.com was created for the MSIT 2k12 batch of ITC Tandojam, and the data from the designed wikispace page was analyzed for two semesters, from 01-Jan 2013 to 31-Dec 2013.This study focused on the reuse of web pages and the data for the next batch of MSIT as an evolution phase. As a result of the need for an archive of the MSIT page, a visual SVN server was also used to provide version control users with remote and local access to the manually stored archives of Wikispaces.The Tortoise SVN tool was used as an SVN client. The Visual SVN server provides the interface between the SVN client and archives of wiki space. Then, using different actions of the Tortoise SVN tool, I analyzed wiki data from a visual SVN server, and revisions were automatically updated on the server.

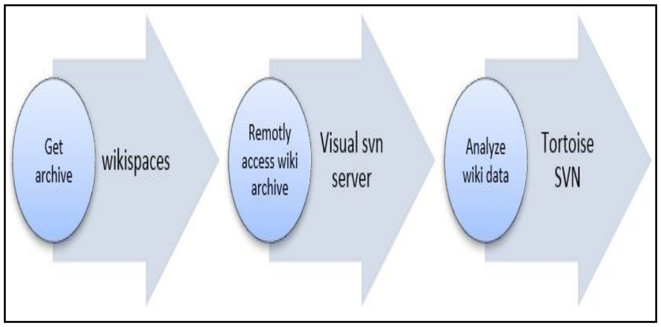


Figure 3.1: Conceptual diagram for SVN to wiki relation

3.2 VISUAL SVN SERVER

Visual SVN Server provides the facility to manage and install a multi-functional subversion server easily on the Windows platform. It is allocated as a separate installation set with the latest versions of all the required components. Visual SVN Server uses the built-in isolated violations over HTTP to communicate with clients and web browsers Apache HTTP Server.

3.3 TORTOISE SVN CLIENT

Tortoise SVN is free software used by developers to manage different versions of the source code. THE Tortoise SVN tool is used to maintain the SVN repository by applying its multiple functions and commands. It manages the projects that are in a subversion repository. The Tortoise SVN repository has three types of sessions, which are discussed in detail below: tang, branches, and trunk.

3.4 TAG

Tags are used to highlight versions in the repository history. Before releasing any version, the tag must first be created [18].

3.4 BRANCHES

Branches are used to create the development lines for multi-versions of the same product, such as having a storage place to fix bugs in the stable release. is also provided side-line development [19].

3.3 TRUNK

Trunks are the core leading line of development in an SVN repository [20].

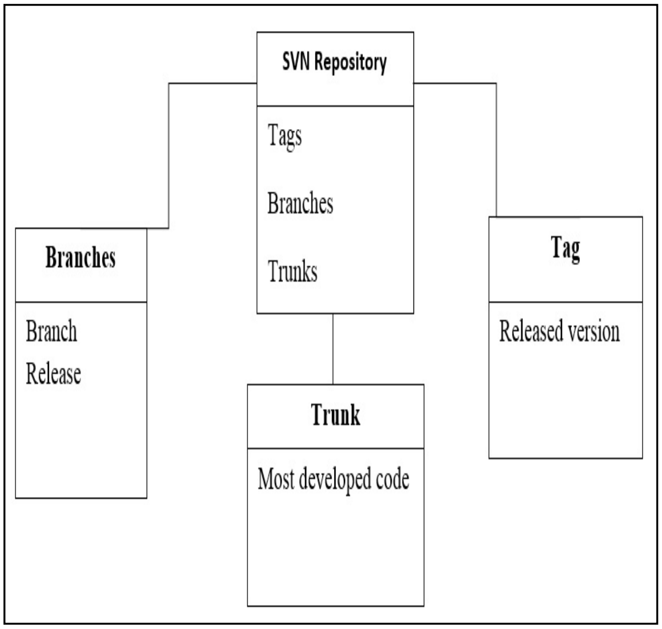


Figure 3.2: SVN repository schemas

3.4 DATA COLLECTION

In this research study, an archive of two semester's academic data which include web pages, lectures, and notifications. It was downloaded from the Master of Science in Information Technology.

3.5 DEPLOYING THE TORTOISE SVN TOOL

Once the archives of the Wikispaces were obtained, the TortoiseSVN client was deployed on multiple machines. The client tool helps in the analysis of SVN by applying different actions such as commit, update, check out, edit, import, and export, etc. shows graphically and statistically output of the analyzed data according to changes that occurred in the latest revisions.

3.6 ANALYZE THE SVN REPOSITORY

So many file folders, all types of data are imported into the repository. The tool facilitates so many action options for edit, remove, update commit, checkout, repo browser, show log, export and so on. The edit option was mostly used for data analysis, and revision numbers increased whenever you edited any file, as all files and pages information were displayed in the repo browser option.In this research, about fifteen (15) users in the form of authors have analyzed data in different ways, dates and times. The Tortoise SVN tool also provided graphical information about revisions, authors and also generated general statistics reports about revisions and authors [21-31].

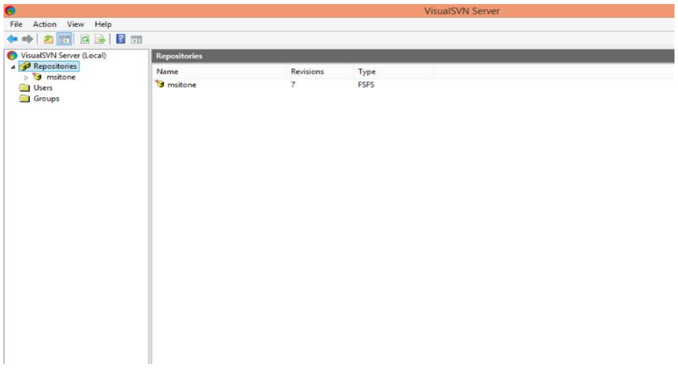
4. RESULTS& DISCUSSION

The results of the proposed research have been observed by using Visual SVN server and Tortoise SVN client. The SVN repository was analyzed by the client and server tools by applying different methods and commands. The working process was carried out in the following steps.

1. Using the SVN server, populate the wiki space data archives.
2. Create user/client for remote access.
3. Accessing repository through Tortoise SVN Tool
4. Checkout repository
5. Analyze files, directories from various aspects.
6. Graphically and statistically results of revisions and commits of authors.

USING THE SVN SERVER, POPULATE THE WIKI SPACE DATA ARCHIVES

Because we needed remote access to our data, we collected MSIT-PI Wiki space information in the form of an Archive and manually inserted the entire data into Visual SVN Server.Wiki data managed in the Visual SVN Server tool is shown in Figure 4.1.

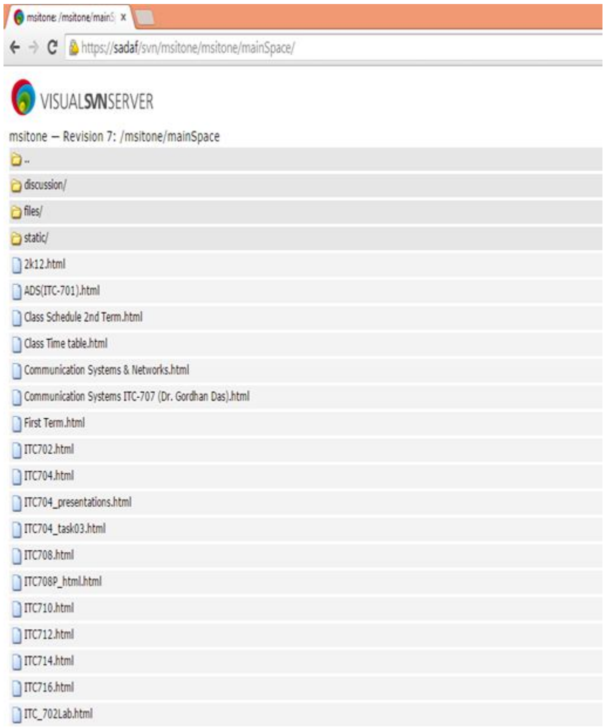


Name	Revisions	Type
msitone	7	FSFS

Figure 4.1: Insert manually data into Visual SVN Server

4.2CREATE USER/CLIENT FOR REMOTE ACCESS

Visual SVN Server was not allowed to access your data without a user account. Create users and then easily access the wiki spaces information remotely. Through this https://sadaf/svn/msitone/ and username: Sadaf password:yyyyy.



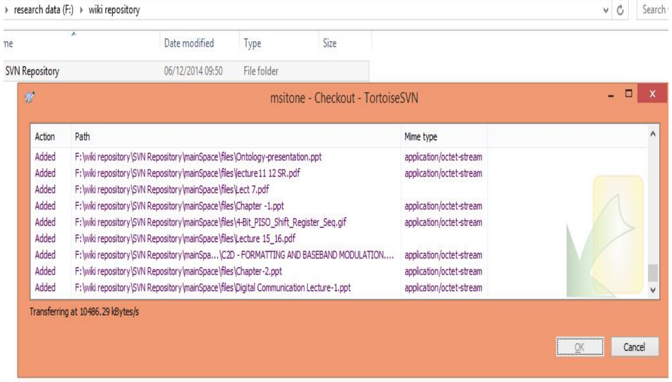
msitone – Revision 7: /msitone/mainSpace

..
discussion/
files/
static/
2k12.html
ADS(ITC-701).html
Class Schedule 2nd Term.html
Class Time table.html
Communication Systems & Networks.html
Communication Systems ITC-707 (Dr. Gordhan Das).html
First Term.html
ITC702.html
ITC704.html
ITC704_presentations.html
ITC704_task03.html
ITC708.html
ITC708P_html.html
ITC710.html
ITC712.html
ITC714.html
ITC716.html
ITC_702Lab.html

Figure 4.2: Remotely Access data from the server

4.3 ACCESSING REPOSITORY THROUGH TORTOISE SVN TOOL

Tortoise SVN Client provides the facility to create a repository for historical data. The process was starting to create an SVN repository for analyzing that Wikispaces information and increase revisions by different authors.


research data (F:) ▸ wiki repository
Search
msitone - Checkout - TortoiseSVN
OK
Cancel

ne	Date modified	Type	Size
SVN Repository	06/12/2014 09:50	File folder	

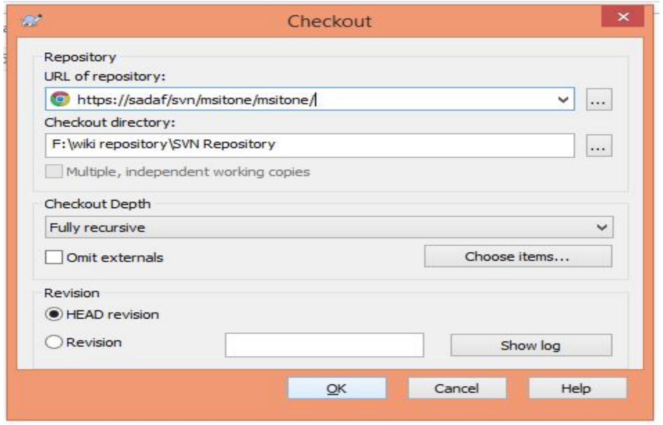
Action	Path	Mime type
Added	F:\wiki repository\SVN Repository\mainSpace\files\Ontology-presentation.ppt	application/octet-stream
Added	F:\wiki repository\SVN Repository\mainSpace\files\lecture11 12 SR.pdf	application/octet-stream
Added	F:\wiki repository\SVN Repository\mainSpace\files\Lect 7.pdf	
Added	F:\wiki repository\SVN Repository\mainSpace\files\Chapter -1.ppt	application/octet-stream
Added	F:\wiki repository\SVN Repository\mainSpace\files\4-Bit_PISO_Shift_Register_Seq.gif	application/octet-stream
Added	F:\wiki repository\SVN Repository\mainSpace\files\Lecture 15_16.pdf	
Added	F:\wiki repository\SVN Repository\mainSpa...\C2D - FORMATTING AND BASEBAND MODULATION....	application/octet-stream
Added	F:\wiki repository\SVN Repository\mainSpace\files\Chapter-2.ppt	application/octet-stream
Added	F:\wiki repository\SVN Repository\mainSpace\files\Digital Communication Lecture-1.ppt	application/octet-stream

Transferring at 10486.29 kBytes/s

Figure 4.3: Process of creating SVN Repository By using Tortoise SVN Client Tool.

4.4 CHECKOUT REPOSITORY

Checkout was a very important phase in this whole process. when applied checkout on SVN repository we used URL of data that were accessible through Visual SVN Server and established link/interface between both tools and we easily analyzed SVN repository.



URL of repository: https://sadaf/svn/msitone/msitone/

Checkout directory: F:\wiki repository\SVN Repository

□ Multiple, independent working copies

Fully recursive

□ Omit externals

● HEAD revision

○ Revision

Figure 4.4: Apply check out On SVN Repository and give the server URL.

After that username and password required for Authentication and then started the process of checkout. Like that,

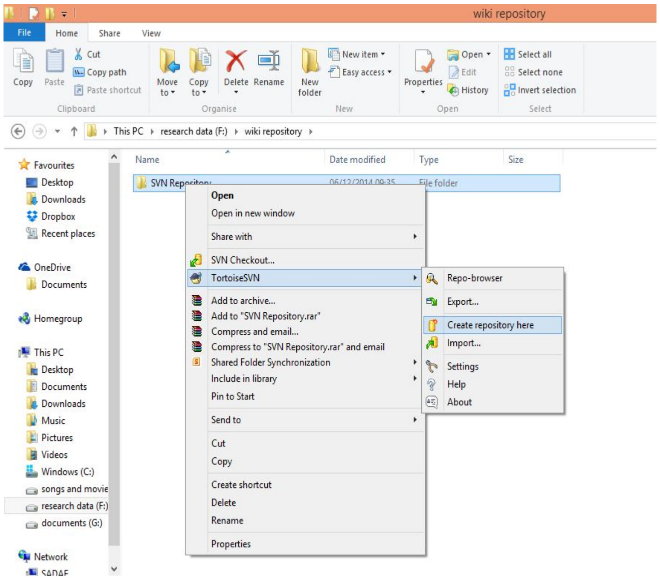
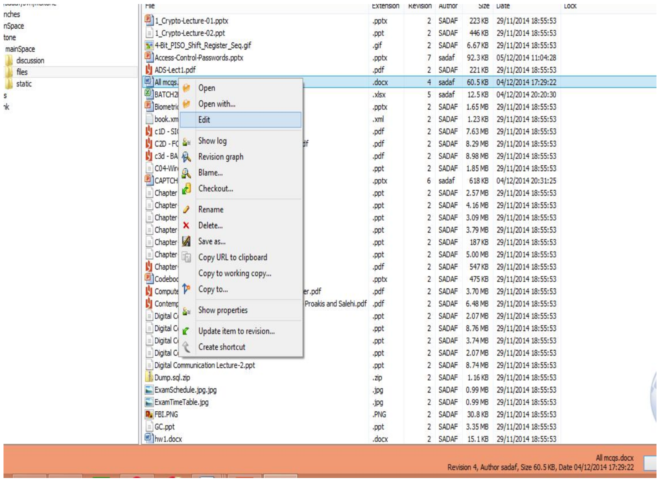


Figure 4.5: Show after the authentication process of checkout data from the server

4.5 ANALYZE FILES, DIRECTORIES THROUGH DIFFERENT ACTIONS

The analyzing phase was started; the Go repo browser and then applied more actions of the Tortoise SVN tool on that data and the important thing was that the changes were automatically updated on the server.



File	Extension	Revision	Author	Size	Date	Lock
1_Crypto-Lecture-01.pptx	.pptx	2	SADAF	223 KB	29/11/2014 18:55:53	
1_Crypto-Lecture-02.ppt	.ppt	2	SADAF	446 KB	29/11/2014 18:55:53	
4-Bit_PISO_Shift_Register_Seq.gif	.gif	2	SADAF	6.67 KB	29/11/2014 18:55:53	
Access-Control-Passwords.pptx	.pptx	7	sadaf	92.3 KB	05/12/2014 11:04:28	
ADS-Lect1.pdf	.pdf	2	SADAF	221 KB	29/11/2014 18:55:53	
All mcqs.	.docx	4	sadaf	60.5 KB	04/12/2014 17:29:22	
BATCH2	.xlsx	5	sadaf	12.5 KB	04/12/2014 20:20:30	
Biometric	.pptx	2	SADAF	1.65 MB	29/11/2014 18:55:53	
book.xm	.xml	2	SADAF	1.23 KB	29/11/2014 18:55:53	
c1D - SI	.pdf	2	SADAF	7.63 MB	29/11/2014 18:55:53	
C2D - FC df	.pdf	2	SADAF	8.29 MB	29/11/2014 18:55:53	
c3d - BA	.pdf	2	SADAF	8.98 MB	29/11/2014 18:55:53	
C04-Wir	.ppt	2	SADAF	1.85 MB	29/11/2014 18:55:53	
CAPTCH	.pptx	6	sadaf	618 KB	04/12/2014 20:31:25	
Chapter	.ppt	2	SADAF	2.57 MB	29/11/2014 18:55:53	
Chapter	.ppt	2	SADAF	4.16 MB	29/11/2014 18:55:53	
Chapter	.ppt	2	SADAF	3.09 MB	29/11/2014 18:55:53	
Chapter	.ppt	2	SADAF	3.79 MB	29/11/2014 18:55:53	
Chapter	.ppt	2	SADAF	187 KB	29/11/2014 18:55:53	
Chapter	.ppt	2	SADAF	5.00 MB	29/11/2014 18:55:53	
Chapter	.pdf	2	SADAF	547 KB	29/11/2014 18:55:53	
Codeboo	.pptx	2	SADAF	475 KB	29/11/2014 18:55:53	
Compute er.pdf	.pdf	2	SADAF	3.70 MB	29/11/2014 18:55:53	
Contemp Proakis and Salehi.pdf	.pdf	2	SADAF	6.48 MB	29/11/2014 18:55:53	
Digital C	.ppt	2	SADAF	2.07 MB	29/11/2014 18:55:53	
Digital C	.ppt	2	SADAF	8.76 MB	29/11/2014 18:55:53	
Digital C	.ppt	2	SADAF	3.74 MB	29/11/2014 18:55:53	
Digital C	.ppt	2	SADAF	2.07 MB	29/11/2014 18:55:53	
Digital Communication Lecture-2.ppt	.ppt	2	SADAF	8.74 MB	29/11/2014 18:55:53	
Dump.sql.zip	.zip	2	SADAF	1.16 KB	29/11/2014 18:55:53	
ExamSchedule.jpg.jpg	.jpg	2	SADAF	0.99 MB	29/11/2014 18:55:53	
ExamTimeTable.jpg	.jpg	2	SADAF	0.99 MB	29/11/2014 18:55:53	
FBI.PNG	.PNG	2	SADAF	30.8 KB	29/11/2014 18:55:53	
GC.ppt	.ppt	2	SADAF	3.35 MB	29/11/2014 18:55:53	
hw1.docx	.docx	2	SADAF	15.1 KB	29/11/2014 18:55:53	

All mcqs.docx
Revision 4, Author sadaf, Size 60.5 KB, Date 04/12/2014 17:29:22

Figure 4.6: Applying different actions of Tortoise SVN Client tool

4.6 GRAPHICALLY AND STATISTICALLY RESULTS OF REVISIONS AND COMMITS BY AUTHORS

Tortoise SVN client provides graphically and statistically results of all activities which were applied to that data. Results based on commits, authors, date, and files, etc.

4.6.1 STATISTICALLY RESULTS

Statistical results contain total counts of revisions, commits, files, authors and most, least users. These results are generated with Tortoise SVN. One of the sample results is described in Table 4.1 and 4.2.

Table 4.1: Result of General statistical SVN Repository

First revision number	1
First revision number	**50**
Total file revision count	**247**
Author count	**15**
First revision date	**Sep 22, 2014 06:37pm**
Last revision date	**Nov 26, 2014 09:39am**
Most active author	**Sadaf Solangi**
Least active author	**Search**
Number of weeks count	**4**

Table 4.2: Top 5 files and 10 hot user List

Ranking Number	Authors	Files
1	Sadaf Solangi	All mcqs.docx
2	Forum	BATCH2K12-LIST.xlcx
3	Gulshan	1-crypto-lecture-02.pptx
4	Haseena	4-bit-PISO-Shift-Register-Seq.gif
5	Mehtab	ExamSchedule.jpg.jpg
6	Kareena	
7	Bushra Tahira	
8	Noreen	
9	Paras	
10	Sahni Abbasi	

4.6.2 GRAPHICALLY RESULTS

Graphically results from consists (revisions, commits, authors, date).

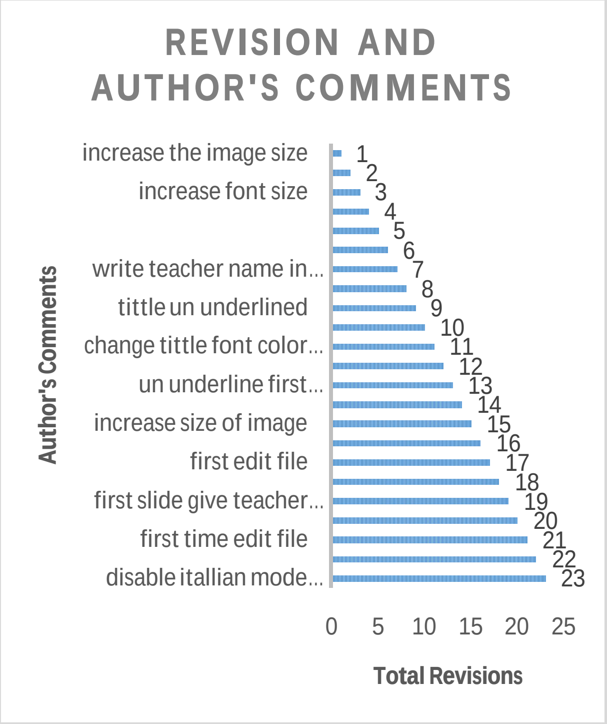


Figure 4.7:Commits on revision and total revisions.

Figure 4.7 Show all of the commits given by authors on different SVN repository files and dates and the total number of revisions was also calculated.

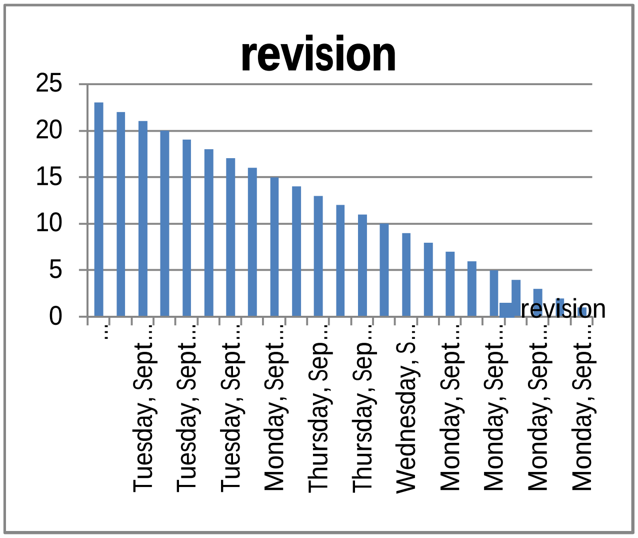


Figure 4.8:Describes total revisions and every revision has been done on multiple dates.

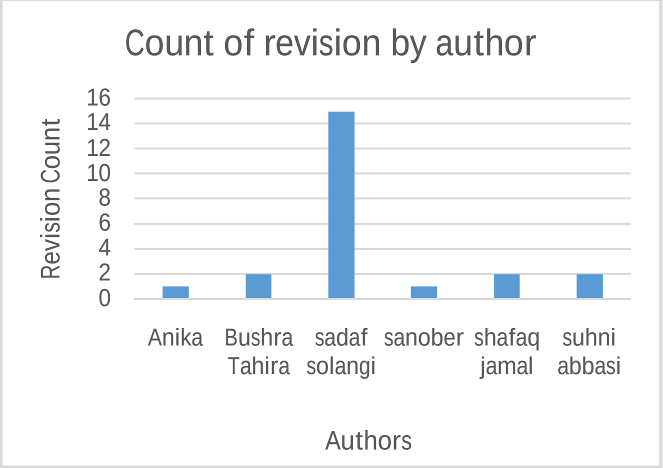


Figure 4.9: Authors and revision count

Figure 4.9 shows the results of different authors and how many percent's every author has been participating in the form m edit any file gives the commits and also applied change delete modify commands.

4.7 DISCUSSION

This research focused on the SVN repository of http://msitone.wikispaces.com data which includes daily lectures, assignments, notifications, and so on. The purpose of this research was to maintain the http://msitone.wikispaces.com repository in a well-defined manner. It was also used to introduce new versions by changing in directories, files, codes, and web pages. New versions were introduced to save the efforts of users by using repositories for new batches and teachers of ITC, SAU Tando jam. In this research difference of using SVN repository and without using the SVN repository was analyzed. SVN repository used server and client-side tools, because of that data could be easily accessed remotely by using URL and automatically update on the server. While without SVN repository new versions could not be introduced. It was also time-consuming for managing the data. It had no access to multiple users. Without repository, one could just manage or edit files locally. It was analyzed from the literature that SVN repositories were introduced for the institutional purpose of managing and remotely accessing. While in this study SVN repository was used for managing, remotely accessing and, re-using older data for the next batch and teachers and also introducing versions by different authors.

The results were supported by Germanin which CVS was used to maintain files for software development. The results had also shown similarity with [8] who used Boa, a domain-specific language, and infrastructure to testing MSR (Mining Software Repositories) which substantially decreases programming efforts. The results were also supported by the [16] that had managed the institutional repository and analyzed that repository was more useful than a regular basis update of any academic record.

5. SUMMARY & CONCLUSION

Based on results obtained from the present study, it is summarized that the study was focused on maintaining, analyzing and reusing the http://msitone.wikispaces.com repository. The data of the academic year 2012-2013 was stored in the form of archives and backups. Reusing that data for future purposes needs extra efforts and resources. To overcome this problem the SVN repository was developed on archives of wiki created for 2K12 master batch of Information Technology Centre. In this research, the archive was downloaded from the MSIT Wikispaces page. For the downloaded archive repository was created on the client-side and then it was stored on the server. Multiple users and authors had accessed and analyzed that repository.In this study, the SVN repository was analyzed by using Visual SVN Server and Tortoise SVN Client tools. SVN Tortoise Client tool was used for maintaining SVN repository by commands like commit, update, checkout, delete, revisions. Visual SVN Server was used for remotely storage purposes.Manually editing of data was time-consuming, if there is no backup of data, one cannot revert the changes at a specific point.

The basis of the presented study, the following conclusion could be drawn:

- It was concluded from the results that by using SVN repositories efforts of users and resources could be saved.
- It was also concluded that using the SVN repository was better than manually updating data.
- If the newer version of the repository was not suitable, the user could revert to the older version.

ACKNOWLEDGEMENT

This work is idea of authors and supervisor. Authors thanks to the reviewers and editor of journal for constructive comments.

REFERENCES

1. Ali, S., S.Jan and I.Amin.2013. Status of Open Access Repositories: A Global Perspective. In Journal,1(1):35-42.
2. Bernstein, P. A. and U. Dayal.1994. An Overview of Repository Technology. Conf VLDB, Pp.705-713.
3. Beyer, D. and A. Noack. 2005. Clustering Software Artifacts Based on Frequent Common Changes. IWPC.1-10.
4. Björk, B.C.2013. Open Access Subject Repositories – an Overview.journal(ASIST), Pp.1-20.
5. Cohen, A., S.Kalimi, And R.Nachmias. 2013. The Use of Digital Repositories for Enhancing Teacher Pedagogical Performance.Interdisciplinary Journal of E-Learning and Learning Objects, 9: 201-218.
6. CollabNet. 2014. CollabNet Subversion. http://www.collab.net/news/press/ameritas-selects-collabnet-teamforge-its-agile-alm-platform-drive-cost-savings-andLast visited on: 3/16/2014
7. Covey, D.T.2011. Recruiting Content for the Institutional Repository: The Barriers Exceed the Benefits. J. Digit. Inf,12(3):1-18.
8. Dyer, R., H. A. Nguyen, H. Rajan, and T.N. Nguyen. 2013. Boa: a language and infrastructure for analyzing ultra-large-scale software repositories. ICSE. pp.422-431.
9. German, D. M. 2004. Mining CVS repositories, the soft change experience: In Proc.Int'l Workshop on Mining Software Repositories, pp.17-21.
10. Giesecke, J.2011. Institutional Repositories: Keys to Success.Journal of Library Administration, 51(5/6):529-542.
11. Group, J.V and C. Prefer. 2006. Version management tools as a basis for integrating Product Derivation and Software Product Families. In Proc of the Workshop on Variability Management-Working with Variability Mechanisms at SPLC, pp. 48-57.
12. Janjic, W., O. Hummel, M. Schumacher, and C. Atkinson. 2013. An unabridged source code dataset for research in software reuse. J. MSR. pp. 339-342.
13. Kankanhalli, A., O.K(D).Lee.and K.H.Lim.2011. Knowledge Reuse through Electronic Repositories: A Study in the Context of Customer Service Support.J.inf-management,48(2-3):106-113.
14. Kiefer, C., A. Bernstein, and J. Tappolet. 2007. Mining Software Repositories with iSPAROL and a Software Evolution Ontology. In Proc. of the 4th Int’l Workshop on Mining Software Repositories.

15. Kim, S., T. Zimmermann, M. Kim, A. Hassan, A. Mockus, T. Girba, M. Pinzger, E. J. Whitehead, Jr, and A. Zeller. 2006. TA-RE. An exchange language for mining software repositories. J. MSR.22-25.
16. Kingsley, D.A.2010. The advocacy and awareness are imperative: a repository overview.conf(VALA).1-13.
17. Klungthanaboon, W., T. Leelanupab and M. Moss.2012. institutional repositories for scholarly communities in Thailand. KMITL Information Technology Journal, Pp.1-14.
18. Kussmaul, C. and M. College. 2008. Supporting teams with open-source software tools. J. NCIIA. Pp.141-147.
19. Mahomes, L.K, and M. Koganuramath.2006. Institutional Repository Of Jawaharlal Nehru University Library, New Delhi.workshop on the development of institutional repository using Dspace(WDIR).64-70.
20. Mitchell, B.T.and M.R. Subramani.2010. Knowledge repositories and knowledgeable action. Conf (ICIS), Pp.195.
21. Nicholas, D., I. Rowlands, A. Watkinson, D. Brown and H. R Jamali. 2012. Digital repositories ten years on: what do scientific researchers think of them and how do they use them? J. Learned Publishing. 25(3): 195–206.
22. Nicholas, D., I.Rowlands., A.Watkinson., D.Brown., B.Russell. and H.R. Jamali.2013. Have digital repositories come of age? The views of library directors. Journal (Webology),10(2):1-16.
23. Rao, P.V.2007. Institutional Repositories: A Key Role For Libraries. In’t CALIBER, Pp.689-695.
24. Shearer, K.2003.institutional repositories: Towards the identification of critical success factors. Conf ACSI, Pp.250-263.
25. Shields, P., N. Rangarajan, and l. Stewart. 2012. Open access digital repository sharing student research with the world. JPAE.18 (1): 157-181.
26. Sowe, S. K., I. Samoladas, I. Stamelos, and L. Angelis. 2008. Are FLOSS developers committing to CVS/SVN as much as they are talking in mailing lists? Challenges for integrating data from Multiple Repositories. J. WoPDaSD. Pp.49-54.
27. Taufer, M., P.J.Teller, A. Kerstens, and R. Romero. 2007. Collaborative Research Tools for Students, Staff, and Faculty.In Proc. of the Int’l SUN Conference on Teaching and Learning. Pp.1-6.
28. Voinea, L and A.Telea. 2006.Mining Software Repositories with CVS grab: In Proc. of the 2006 Int’l Workshop on Mining Software Repositories, pp.167-168.
29. Wilson, M.C and R.C. jantz.2011. Building value-added services for institutional repositories (IRs): Modeling the Rutgers experience. IFLA satellite conference, Pp.1-14.
30. Zimmermann, T., P. Weißgerber, S. Diehl and A. Zeller. 2005.Mining Version Histories to Guide Software Changes. IEEE Transactions on Software Engineering.31 (6):429-445.
31. Zuccala, A., C. Oppenheim. And R. Dhiensa.2008. Managing and evaluating digital repositories.journal(inf.res),13(1):1-32.