INTERNATIONAL JOURNAL OF RESEARCH IN COMMERCE, IT & MANAGEMENT

IJRCM

A Monthly Double-Blind Peer Reviewed (Refereed/Juried) Open Access International e-Journal - Included in the International Serial Directories

Indexed & Listed at:

Ulrich's Periodicals Directory ©, ProQuest, U.S.A., EBSCO Publishing, U.S.A., Cabell's Directories of Publishing Opportunities, U.S.A., Open J-Gage, India [link of the same is duly available at Inflibnet of University Grants Commission (U.G.C.)], Index Copernicus Publishers Panel, Poland with IC Value of 5.09 & number of libraries all around the world.

Circulated all over the world & Google has verified that scholars of more than 1771 Cities in 148 countries/territories are visiting our journal on regular basis.

Ground Floor, Building No. 1041-C-1, Devi Bhawan Bazar, JAGADHRI – 135 003, Yamunanagar, Haryana, INDIA

http://ijrcm.org.in/

CONTENTS

CALL FOR MANUSCRIPTS

We invite unpublished novel, original, empirical and high quality research work pertaining to recent developments & practices in the area of Computer, Business, Finance, Marketing, Human Resource Management, General Management, Banking, Insurance, Corporate Governance and emerging paradigms in allied subjects like Accounting Education; Accounting Information Systems; Accounting Theory & Practice; Auditing; Behavioral Accounting; Behavioral Economics; Corporate Finance; Cost Accounting; Econometrics; Economic Development; Economic History; Financial Institutions & Markets; Financial Services; Fiscal Policy; Government & Non Profit Accounting; Industrial Organization; International Economics & Trade; International Finance; Macro Economics; Micro Economics; Monetary Policy; Portfolio & Security Analysis; Public Policy Economics; Real Estate; Regional Economics; Tax Accounting; Advertising & Promotion Management; Business Education; Management Information Systems (MIS); Business Law, Public Responsibility & Ethics; Communication; Direct Marketing; E-Commerce; Global Business; Health Care Administration; Labor Relations & Human Resource Management; Marketing Research; Marketing Theory & Applications; Non-Profit Organizations; Office Administration/Management; Operations Research/Statistics; Organizational Behavior & Theory; Organizational Development; Production/Operations; Public Administration; Purchasing/Materials Management; Retailing; Sales/Selling; Services; Small Business Entrepreneurship; Strategic Management Policy; Technology/Innovation; Tourism, Hospitality & Leisure; Transportation/Physical Distribution; Algorithms; Artificial Intelligence; Compilers & Translation; Computer Aided Design (CAD); Computer Aided Manufacturing; Computer Graphics; Computer Organization & Architecture; Database Structures & Systems; Digital Logic; Discrete Structures; Internet; Management Information Systems; Modeling & Simulation; Multimedia; Neural Systems/Neural Networks; Numerical Analysis/Scientific Computing; Object Oriented Programming; Operating Systems; Programming Languages; Robotics; Symbolic & Formal Logic and Web Design. The above mentioned tracks are only indicative, and not exhaustive.

Anybody can submit the soft copy of his/her manuscript **anytime** in M.S. Word format after preparing the same as per our submission guidelines duly available on our website under the heading guidelines for submission, at the email address: **infoijrcm@gmail.com**.

GUIDELINES FOR SUBMISSION OF MANUSCRIPT

1. **COVERING LETTER FOR SUBMISSION:**

DATED: _____

THE EDITOR
IJRCM

Subject: **SUBMISSION OF MANUSCRIPT IN THE AREA OF _____.**

(e.g. Finance/Marketing/HRM/General Management/Economics/Psychology/Law/Computer/IT/Engineering/Mathematics/other, please specify)

DEAR SIR/MADAM

Please find my submission of manuscript entitled '_____' for possible publication in your journals.

I hereby affirm that the contents of this manuscript are original. Furthermore, it has neither been published elsewhere in any language fully or partly, nor is it under review for publication elsewhere.

I affirm that all the author (s) have seen and agreed to the submitted version of the manuscript and their inclusion of name (s) as co-author (s).

Also, if my/our manuscript is accepted, I/We agree to comply with the formalities as given on the website of the journal & you are free to publish our contribution in any of your journals.

NAME OF CORRESPONDING AUTHOR:
Designation:
Affiliation with full address, contact numbers & Pin Code:
Residential address with Pin Code:
Mobile Number (s):
Landline Number (s):
E-mail Address:
Alternate E-mail Address:

NOTES:
a) The whole manuscript is required to be in ***ONE MS WORD FILE*** only (pdf. version is liable to be rejected without any consideration), which will start from the covering letter, inside the manuscript.
b) The sender is required to mention the following in the **SUBJECT COLUMN** of the mail:
New Manuscript for Review in the area of (Finance/Marketing/HRM/General Management/Economics/Psychology/Law/Computer/IT/ Engineering/Mathematics/other, please specify)
c) There is no need to give any text in the body of mail, except the cases where the author wishes to give any specific message w.r.t. to the manuscript.
d) The total size of the file containing the manuscript is required to be below **500 KB**.
e) Abstract alone will not be considered for review, and the author is required to submit the complete manuscript in the first instance.
f) The journal gives acknowledgement w.r.t. the receipt of every email and in case of non-receipt of acknowledgment from the journal, w.r.t. the submission of manuscript, within two days of submission, the corresponding author is required to demand for the same by sending separate mail to the journal.

2. **MANUSCRIPT TITLE:** The title of the paper should be in a 12 point Calibri Font. It should be bold typed, centered and fully capitalised.

3. **AUTHOR NAME (S) & AFFILIATIONS:** The author (s) **full name**, **designation**, **affiliation** (s), **address**, **mobile/landline numbers**, and **email/alternate email address** should be in italic & 11-point Calibri Font. It must be centered underneath the title.

4. **ABSTRACT:** Abstract should be in fully italicized text, not exceeding 250 words. The abstract must be informative and explain the background, aims, methods, results & conclusion in a single para. Abbreviations must be mentioned in full.

5. **KEYWORDS:** Abstract must be followed by a list of keywords, subject to the maximum of five. These should be arranged in alphabetic order separated by commas and full stops at the end.

6. **MANUSCRIPT:** Manuscript must be in ***BRITISH ENGLISH*** prepared on a standard A4 size ***PORTRAIT SETTING PAPER***. It must be prepared on a single space and single column with 1" margin set for top, bottom, left and right. It should be typed in 8 point Calibri Font with page numbers at the bottom and centre of every page. It should be free from grammatical, spelling and punctuation errors and must be thoroughly edited.

7. **HEADINGS:** All the headings should be in a 10 point Calibri Font. These must be bold-faced, aligned left and fully capitalised. Leave a blank line before each heading.

8. **SUB-HEADINGS:** All the sub-headings should be in a 8 point Calibri Font. These must be bold-faced, aligned left and fully capitalised.

9. **MAIN TEXT:** The main text should follow the following sequence:

   INTRODUCTION

   REVIEW OF LITERATURE

   NEED/IMPORTANCE OF THE STUDY

   STATEMENT OF THE PROBLEM

   OBJECTIVES

   HYPOTHESES

   RESEARCH METHODOLOGY

   RESULTS & DISCUSSION

   FINDINGS

   RECOMMENDATIONS/SUGGESTIONS

   CONCLUSIONS

   SCOPE FOR FURTHER RESEARCH

   ACKNOWLEDGMENTS

   REFERENCES

   APPENDIX/ANNEXURE

   It should be in a 8 point Calibri Font, single spaced and justified. The manuscript should preferably not exceed ***5000 WORDS***.

10. **FIGURES & TABLES:** These should be simple, crystal clear, centered, separately numbered & self explained, and **titles must be above the table/figure**. **Sources of data should be mentioned below the table/figure**. It should be ensured that the tables/figures are referred to from the main text.

11. **EQUATIONS:** These should be consecutively numbered in parentheses, horizontally centered with equation number placed at the right.

12. **REFERENCES:** The list of all references should be alphabetically arranged. The author (s) should mention only the actually utilised references in the preparation of manuscript and they are supposed to follow **Harvard Style of Referencing**. The author (s) are supposed to follow the references as per the following:

- All works cited in the text (including sources for tables and figures) should be listed alphabetically.
- Use (**ed.**) for one editor, and (**ed.s)** for multiple editors.
- When listing two or more works by one author, use --- (20xx), such as after Kohl (1997), use --- (2001), etc, in chronologically ascending order.
- Indicate (opening and closing) page numbers for articles in journals and for chapters in books.
- The title of books and journals should be in italics. Double quotation marks are used for titles of journal articles, book chapters, dissertations, reports, working papers, unpublished material, etc.
- For titles in a language other than English, provide an English translation in parentheses.
- The location of endnotes within the text should be indicated by superscript numbers.

PLEASE USE THE FOLLOWING FOR STYLE AND PUNCTUATION IN REFERENCES:

BOOKS

- Bowersox, Donald J., Closs, David J., (1996), "Logistical Management." Tata McGraw, Hill, New Delhi.
- Hunker, H.L. and A.J. Wright (1963), "Factors of Industrial Location in Ohio" Ohio State University, Nigeria.

CONTRIBUTIONS TO BOOKS

- Sharma T., Kwatra, G. (2008) Effectiveness of Social Advertising: A Study of Selected Campaigns, Corporate Social Responsibility, Edited by David Crowther & Nicholas Capaldi, Ashgate Research Companion to Corporate Social Responsibility, Chapter 15, pp 287-303.

JOURNAL AND OTHER ARTICLES

- Schemenner, R.W., Huber, J.C. and Cook, R.L. (1987), "Geographic Differences and the Location of New Manufacturing Facilities," Journal of Urban Economics, Vol. 21, No. 1, pp. 83-104.

CONFERENCE PAPERS

- Garg, Sambhav (2011): "Business Ethics" Paper presented at the Annual International Conference for the All India Management Association, New Delhi, India, 19–22 June.

UNPUBLISHED DISSERTATIONS AND THESES

- Kumar S. (2011): "Customer Value: A Comparative Study of Rural and Urban Customers," Thesis, Kurukshetra University, Kurukshetra.

ONLINE RESOURCES

- Always indicate the date that the source was accessed, as online resources are frequently updated or removed.

WEBSITES

- Garg, Bhavet (2011): Towards a New Natural Gas Policy, Political Weekly, Viewed on January 01, 2012 http://epw.in/user/viewabstract.jsp

QUALITY IMPROVEMENT IN FREE AND OPEN SOURCE SOFTWARE PROJECTS

DR. SHAIK MAHABOOB BASHA
LECTURER
DEPARTMENT OF COMPUTER SCIENCE
GOVERNMENT DEGREE COLLEGE
NANDIKOTKUR

ABSTRACT

Free and Open Source software has had major impact on the computer industry. Free and Open Source software or FOSS allows the different groups to share their source code. Open source projects are developed by students as well as developers. Some of the best examples are MySQL, relational database, the Apache Web Server and the Sendmail mail transport agent. It has made life easy enabling us to develop applications pertaining to our requirements. We can use other applications by modifying their basic codes, thus enabling power into our hands. There is a scope in developing applications based on cyber security. It is easier and faster to process information. The sudden success and major adoption of this new and innovative software development strategy has raised many questions, attracted the interest of academics in a variety of disciplines and promoted interdisciplinary research. Licensed software guarantees unrestricted use, access to the source code, and the right to modify and to distribute source code. Free Software is associated with a strong philosophical focus on freedom, adherents of the open source movement stress features of the software, such as high quality. FOSS is now used in many areas, such as the Internet.

KEYWORDS
FOSS, MySQL, source code, Apache, Web Server, Send mail.

INTRODUCTION
FOSS is a combination of ideas of communities which cooperatively develop and share software and its source code can be traced back several decades to the origin of the modern computers. FOSS is a license model for software distribution. Unlike traditional software, this is inherently associated with the protection of proprietary rights. FOSS encompasses a class of licenses which give the user a number of rights and permissions. A FOSS development tool has led to the production of software of significant quality and functionality. There is growing interest in understanding and utilizing the process underlying FOSS development. Various FOSS applications such as Apache and Linux have attained major market share and often dominate their product categories. Major players in the IT sector such as IBM & HP have invested billions of dollars in FOSS development. Furthermore companies like Red Hat which commercialize Linux have attained considerable profit. The collaborative innovation process employed by FOSS projects is often seen as a major organizational innovation. In recent years new attention has been given in the literature to the identification of new processes in software production and other areas of technology that allow organizations to cope with rapid change

QUALITY IMPROVEMENT IN FOSS PROJECTS
The measurement of quality cannot directly be applied to FOSS projects which usually do not have an explicit design or written specifications. Nevertheless, the idea of fitness for purpose applies to any product including FOSS. Many different factors have to be taken into consideration for quality. After all, users have to download and install FOSS themselves and they will usually choose software which fulfils their needs and quality expectations. There are many approaches to quality improvement. The predominant view of quality improvement is process thinking. A high quality product can be created through a high quality process. Quality and productivity increase as 'process variability' Quality improvement can only be achieved in a company through the participation of everyone in the workforce. This view is the compatible with process improvement which takes all aspects of the process into account, including developers and other people participating in the development process. Process improvement view with the aim to find problems with processes employed by FOSS projects to identify ways to improve there process and to find best practices. There are many FOSS projects which have attained significant popularity and quality. There is also growing literature that empirically backs up the anecdotal evidence that FOSS is of higher than expected and there are other empirical studies which show that FOSS is comparable or better than proprietary, closed source software with regards to various quality factors. However these studies have to be regarded with some caution. First, it is not clear what software could act as a basis for comparison. Second, it is easy to misinterpret a given indicator. For example, a high number of reported defects can either indicate buggy and low quality software, but is could also mean that the software is rigorously tested. Finally code quality is only one measure of quality and there are other areas in which FOSS projects do not outperform closed source applications, such as usability.

QUALITY PROBLEMS
The qualities problems that have been identified are yet to be solved in the FOSS community are

1) Unsupported code : one of the unsolved problems is how to handle code that has previously been contributed but which is now unmaintained. A contributor might submit source code to implement a specific feature to an obscure hardware architecture. As changes are made by other developer, this particular feature has to be updated so that is will continue to work. Unfortunately, some of the original contributors may disappear and the code is left unmaintained and unsupported. Lead developers face the difficult decision of how to handle this situation.
2) Configuration management: many FOSS projects offer a high level of customization. While this gives users much flexibility, it also creates problems with testing. It is very difficult or impossible for the lead developer to test all combinations so only the most popular configurations tend to be tested. It is quite common that, when a new release is made, users report that the new version broke their configuration.
3) Security updates : updates in many cases they are made in a timely manner but some
4) Users do not know to report bugs as more users with few technical skills use FOSS, developers see an increase in useless bug reports. In many cases users do not include enough information in a bug report or they file duplicate bug reports. Such reports take unnecessary time away from actual development work. Some projects have tried to write better documentation about reporting bugs but they found that users often do not read the instructions before reporting a bug.
5) Attracting volunteers: a problem some projects face, especially those that are not very popular, is attracting volunteers. There are usually many ways of contributing to a project, such as coding, testing or triaging bugs. However, many projects only find prospective members who are interested in developing new features. Few contributors are interested in helping with testing or triaging bugs. As a result, developers have to use a large portion of their time for tasks other people could easily handle.
6) Lack of documentation: it is possible that the previous problem is related to the lack of documentation. Volunteers may want to contribute in an area but they might not know how to start. Little help is given to prospective contributors and almost no documentation exists. The lack of developer documentation also implies that there is no assurance that everyone follows the same techniques and procedures.
7) Problems with coordination and communication: in some projects, there are problems with coordination and communication which can have a negative impact on project quality. Sometimes it is not clear who is responsible for a particular area and therefore bugs cannot be communicated properly. There may also be duplication of efforts and a lack of coordination related to the removal of critical bugs.

PROBLEMS WITH THE FOSS

The different problems identified from the traditional development which are most commonly cited

- FOSS systems are built by potentially large numbers of volunteers
- Work is not assigned; people undertake the work they choose to undertake.
- There is no explicit system-level design, or even detailed design.
- There is no project plan, schedule or list of deliverables.

RELEASE MANAGEMENT IN FOSS

Release management in FOSS is an area which is largely unexplored. FOSS is characterized by a highly iterative development model in which new development releases are made available very frequently. The aim of the release approach is to gather feedback early and it allows the community to influence the direction of a project. The three types which have been identified are: development releases aimed at developers interested in working on the project or experienced users who need cutting edge technology major user releases based on a stabilized development tree these releases deliver significant new features and functionality as well as bug fixes to end user and are generally well tested minor releases as updates to existing user releases for example to address security issues or critical defects since developers are experts development releases do not have to be polished and are therefore relatively easy to prepare minor updates to stable releases also require little work since they usually only consist of one or two fixes for security or critical bugs on the other hand a new major user release requires significant efforts the soft ware needs to be thoroughly tested ,various quality assurance tasks have to be performed ,documentation has to be written and the software needs to be packaged up in terms of release authority, it can be observed that major new user releases are typically performed by the project leader or a dedicated release manager where as development and minor user releases can also be prepared by a core member of the development team this again shows the significance that user releases have

Project	*Version control system*
GCC	*SVN*
GNOME	*CVS, SVN*
Nano	*CVS*
Synaptic	*Bzr*
X.org	*git, SVN*

Version Control Systems used by some Projects

PREPARATION OF STABLE RELEASES

The act of preparing a stable release for end-users is a complex set of tasks in which all developers of a project have to work together to deliver a high Quality product. While thee specific release approach may differ from project to project a common pattern has been identified: staged progress towards a release where each stage is associated with increasing levels of control over the changes that are permitted. These control mechanisms are usually known as freezes since the development is slowly halted and eventually bought to a standstill.

The role of the release manager is diverse and demanding, because they have to interact with a large number of different people, understand technical issues but also know how to plan and coordinate. The following taxonomy of skills and characteristics which release managers need has been developed.

1. Community building: showing people that their input is useful. Release managers also need respect in the community in order to perform their work.
2. Strong vision: showing developers in which direction the project should be moving.
3. Discipline: saying 'no' Release manager have to focus on an overall goal and can not make everyone happy.
4. Judgment: gauging the risk and possible impact of a particular change.
5. Attention to detail walking through every lime of code that has changed.
6. Good communication: writing release notes, asking for feedback, interacting with users.
7. Management skills: talking to people, organizing, planning, making sure that all the little things happen.

COMPARISION OF PROPRIETARY SOFTWARE AND FOSS

Quality in either FOSS or closed proprietary software is necessarily higher. However many participants felt that FOSS had a higher potential to achieve greater quality and can react faster to critical issues such as security bugs. There are various reasons for this

1) Its open nature promotes more feedback, which can be used to improve the software. Feedback can either be given in the form of bug reports or feature requests.
2) Motivation was higher in FOSS projects because volunteers could work on whatever they wanted. Open collaboration with other developers and input from others also increase the motivation to work on a piece of software. It is found that this increased motivation had positive effects on the quality of the software.
3) FOSS could attract better human resources because of its distributed nature. FOSS can benefit from a wider range of expertise and knowledge than a traditional software company can usually bring to bear on a problem. A downside of community projects compared to commercial development was the lack of resources and infrastructure.
4) It is very hard to compare open and closed development models because of the opposing philosophy of these models. Since closed source companies often hide their defects and source code, it is hard to make a comparison.

DEVELOPMENT AND QUALITY PRACTICES

One of the surprising insights gained was how greatly development practices and processes differed across projects. The identified practices can be categorized broadly into the following three areas

INFRASTRUCTURE

FOSS projects rely heavily on infrastructure that allows distributed development and collaboration. The important parts are

1) Bug Tracking System: These are used to capture feedback from users. They are often used to store both actual bug reports as well as feature requests
2) Version Control Systems: - These allow multiple people to work on the same code base concurrently and keep track of who makes which changes.
3) Automatic Builds: - These make sure that the newest code in the version control system still builds. The test builds can be done on a number of different hardware or software environments.
4) Mailing Lists: - Used for communication, both between developers and users.

PROCESSES

Foss development follows many processes but a large number of them are not documented anywhere – developers adhere to them implicitly.

1) Joining: projects are require prospective members to follow specific, mostly undocumented, procedures in order to join a project. These procedures vary considerably across projects.

2) Release: different release policies are employed by projects but many follow freeze stages. A feature freeze is the point when no new features are to be incorporated into the code base but there is sufficient time to fix bugs.
3) Branches: these are used to differentiate between versions of a program, for example by having a stable and a development branch. New development tools which make branches easier to deal with, such as Arch, have had a significant impact on the development process.
4) Peer review: typically changes made to the version control system are reviewed by members of the projects, through in most cases; this form of peer review is not very well formalized. Developers hoe that other project participants will lock at their changes but there is often no assurance that this is the case in reality.
5) Testing: in order to ensure that a new release fulfils the standards of a project and that it has no major regressions i.e. bugs that affect functionality that previously worked), some projects have testing check lists. These check lists contain the most important functions and briefly describe how they can be tested. A release is only made when testers on different platforms have gone through the check list and have confirmed that the new version does not display major show-stoppers or regressions.
6) Quality assurance: some organize bug days or bug squashing parties to triage their outstanding bugs. During this work, duplicate bug reports are marked as such, old bugs are reproduced, and bugs are also fixed.

DOCUMENTATION

Contributed source code was checked by the lead developer of a project and then rejected because it did not confirm to the coding style, a style that was not clearly documented anywhere. However, some projects, mainly those that have a large number of contributors, have good documentation.

1) Coding styles: documentation aimed at developers which describes the style which should be used for the source code.
2) Code commit: documentation which describes when and who can make changes in project's version control system.

CONCLUSION

A FOSS project has been substantial development in a number of areas. Since large corporations and others increasingly depend on FOSS. FOSS production has traditionally been perceived as unstructured and unorganized and the majority of FOSS projects consist of voluntary contributors. Corporations can rely on the output of FOSS projects based on quality and study aspects of quality improvement and find ways to ensure high levels of quality in the output of FOSS projects. Release management has been chosen as the specific focus. Time based release strategy as a novel concept of release management worth investigation-in-depth. In contrast to traditional software development which is feature-driven, the goal of time based release management is to produce high quality release according to specific release interval. It is found that feature based release management in FOSS projects is often associated with lack of planning, which leads to problems, such as delays and how levels of quality. Since the FOSS projects consist of volunteers, it is difficult to perform planning because there is no guarantee that features will be ready in time for the next release.

FOSS methodology has been used to develop a significant body of software some with high quality, there have been major changes in the expectations of FOSS in recent years. FOSS is no longer seen as hobby projects done for fun, but significant economic factors are involved now. This increased economic interest in FOSS with millions of users and thousands of organizations relying on this collaboratively developed software has led to new requirements, such as the need for sustainability and reliance

REFERENCES

1) Aberdour ,M.(2007).achieving quality in open source software.IEEE software
2) Amor, J.J., J.M.Gonzalez-Barahona, G. Robles, and I. herraiz (2005).
3) Baetjer, H. (1997). Software as Capital. Los Alamitos, CA: IEEE Computer Society Press.
4) Baldwin, C.Y. and K.b Clark (1997). Managing in an age of modularity. Harvard Bussiness Review 75(5), 84-93.
5) Benkler, Y. (2002). Coase's penguin, or, Linux and the nature of the firm.
6) Cofin J (2006) : Analysis of open source principles in diverse collaborative communities
7) Daniel, S. L and K J Steward (2005). Cohension, structure and software quality observation. In E.Damiani, B. Fitzerald, W.Scacchi, M. Scott and G.Succi (Eds), Open Source Systems, Volume 203
8) Eisenhardt, K.M. (1989). Building theories from case study research Academy of Management Review 14(4)
9) Gacek, C and B. Arief (2004). The many meanings of open source, IEEE software 21(1), 34-40
10) Halloran, T.J. and W.L. Scherlis (2002). High quality and open source software practices. In proceedings of the 2nd workshop on Oen Source Software Engineering, Orlando, FL, USA, ICSE.
11) ISO (1991). International Standard ISO/IEC 9126, Information Technology – Software product evaluation – Quality characteristics and guidelines for their use. Geneva : International Electro technical Commission.
12) Johnson, K (2001). A descriptive process model for open-source software development. Master's thesis, Department of Computer Science, University of Calgary.
13) Kelty, C. (2005). Free science. In J. Feller, B. Fitzgerald, S.A. Hissam
14) Measuring libre software using Debian 3.1 (Sarge) as a case study. Upgrade Magazine.
15) Yale Law journal 112(3), 369-446.

REQUEST FOR FEEDBACK

Dear Readers

At the very outset, International Journal of Research in Commerce, IT and Management (IJRCM) acknowledges & appreciates your efforts in showing interest in our present issue under your kind perusal.

I would like to request you to supply your critical comments and suggestions about the material published in this issue as well as on the journal as a whole, on our E-mail i.e. **infoijrcm@gmail.com** for further improvements in the interest of research.

If you have any queries please feel free to contact us on our E-mail infoijrcm@gmail.com.

I am sure that your feedback and deliberations would make future issues better – a result of our joint effort.

Looking forward an appropriate consideration.

With sincere regards

Thanking you profoundly

Academically yours

Sd/-

Co-ordinator

ABOUT THE JOURNAL

In this age of Commerce, Economics, Computer, I.T. & Management and cut throat competition, a group of intellectuals felt the need to have some platform, where young and budding managers and academicians could express their views and discuss the problems among their peers. This journal was conceived with this noble intention in view. This journal has been introduced to give an opportunity for expressing refined and innovative ideas in this field. It is our humble endeavour to provide a springboard to the upcoming specialists and give a chance to know about the latest in the sphere of research and knowledge. We have taken a small step and we hope that with the active co-operation of like-minded scholars, we shall be able to serve the society with our humble efforts.

Our Other Journals