“With unwearied fingers drawing out the lines of life from living knowledge hid.

- Edmund Spencer
3 - Review of Literature:

This chapter presents a review of some of the contributions, which have been made to the understanding of open source software development.

3.1 - Eric S. Raymond “The Cathedral and the Bazaar”:

Eric S. Raymond (1999) has been active in the open source movement since the early days of the Free Software Foundation in the mid 1980’ies, where Eric S. Raymond wrote different modes for the Emacs editor. Emacs is a highly programmable text editor, often cited as the king of all editors. In fact Emacs was the first software written by Richard M. Stallman, when he started the Free Software Foundation.

The Cathedral and the Bazaar gained instant fame, when Netscape lost the browser war to Microsoft Internet Explorer at the end of 1997. Netscape collapsed and the management was looking for alternative ways to either reconstruct the company or to continue the life of the once successful Netscape browser. On January 23rd, 1998, Netscape made the announcement that they would give away the source code for the next generation of their Netscape Communicator suite. The idea for this move was attributed to the paper “The Cathedral and the Bazaar”. The Cathedral and the Bazaar had been the basis for an internal Netscape document outlining how to give away the source code. Later in the same year a meeting was held between the Netscape executives and some of the more prominent figures of the open source community including Linus Torvalds and Eric S. Raymond. The meeting, among others, discussed licensing problems and what the open source community was interested in.

With much publicity the Netscape source code was released in the 31st March 1998 during an event formed as a synchronous party taking place at several locations around the world and transmitted live on the Internet. The code released from Netscape needed a place to live, where developers could collaborate and consequently Mozilla.org was created. Mozilla is a web site, where developers can interact, get the latest version of the source code and also submit their changed source code.
Undoubtedly, the publicity of Netscape using The Cathedral and the Bazaar as a basis for their decisions has helped both the paper and Eric S. Raymond gaining a high public profile.

The Cathedral and Bazaar has been the subject and point of departure of many a discussion debates and deliberations which have had a profound impact on the library fraternity particularly on the views of Eric S. Raymond as visible from the extent of adoption. The impact of these writings is also seen through several authors in the relating to the local linux groups. Skåne Sjælland Linux User Group (SSLUG) is a 6000+ member user group and the arguments and rhetoric of the writings have been repeated in many forms. However, the paper has also been the subject of severe critique from, among others, Nikolai Bezurokov (1999).

The Cathedral and the Bazaar has influence the mind set of people and acted as the catalyst for the paradigm shift in favor of OSS because their paper was the pioneering input exploring the understanding of OSS. Never the less it is not untrue to say that the Cathedral and the Bazaar contribution have been influence (may be tainted) by the contribution of Eric S. Raymond’s writings. When the project began in late 1998, the effect of Netscape using the paper was still profound.

The Cathedral and the Bazaar is the first and perhaps because of this as mentioned above it became a very influential paper written about open source software development. The Cathedral and the Bazaar tries to explain why open source software development is so successful. In The Cathedral and the Bazaar, Eric S. Raymond uses his personal experience from the fetch mail project to illustrate no less than 19 points.

Eric S. Raymond was impressed that open source development was able to actually develop good stable operating systems. Creating operating systems or other complex software for that sake was thought to require some kind of centralised development, where a selected few knew the system and were allowed to work on the system. These systems grew in the hands of their creators with long periods of time between new versions. A great deal of this time was devoted to fixing bugs in the software and creating high quality software with few bugs, before each new version was released.
Open source software development as exemplified by Linux illustrated a different approach to developing software.

According to the Cathedral and Bazaar there are several reasons for the success of open source software. The points and reasons related to open source software development are interesting and explain some of the mechanisms believed to be present in open source software development.

The motivation for participating in open source software development and starting an open source software project is explained with “Every good work of software starts by scratching the developer’s personal itch”. If a person needs a program and the program is not available, the person will soon find himself developing the needed software – provided he possesses the required skills.

In the process of developing open source software the Cathedral and Bazaar suggests that the maintainer must treat users of a project as co-developers. Treating users as codevelopers means listening to the users and using their suggestions, in effect making them part of the project development. This produces interested users, who feel responsibility for the project and this is a motivating factor, which will improve the quality of the code and the speed of development.

In open source software development it is important that the project shows a plausible promise. That is, the software may be crude and buggy and not functioning well, but it still shows promise. Eric S. Raymond notes: “What it must not fail to do is (a) run and (b) convince potential co-developers that it can be evolved into something really neat in a foreseeable future.

3.2 - Eric S. Raymond “Homesteading the Noosphere”:

Homesteading the Noosphere also by Eric S. Raymond is more theoretical and focuses on what mechanisms make open source software development work. The Lockean theory of land tenure is shown to match the behaviour of open source developers in the situation of acquiring projects. Attached to this is the value perspective that is introduced to explain the incentives of open source developers.
this view the incentives of open source developers are based on reputation that can be gained from giving code (gifts) to the community. The community is a surplus community and thus standard economics do not apply and therefore gifts are the real currency. However, one important resource is scarce: Time. Compared to the work that needs to be done, developers only have very limited time at their hands. Whereas the paper goes a long way in explaining the mechanisms related to so-called ownership of open source software projects, the reputation/gift incentive seems simplified. The ideas presented in the Cathedral and Bazaar and Homesteading the Noosphere are recognised in all other literature reviewed

3.3 - The Simple Economics of Open Source Software:

The Simple Economics of Open Source by Lerner and Tirole is a compelling account that contributes to the understanding of both coordination of open source software projects and developers’ motivation. With regard to coordination Lerner & Tirole has a rather rigid understanding of how open source software projects work.

In a paper (Edwards 2000) published in FirstMonday discussed the very perception of leadership and how leaders lead open source software projects presented by Lerner & Tirole. The details of the discussion can be found in the mentioned paper. However, some points from the critique are worth mentioning. It is important to understand that leaders in open source software projects have little means of directing the efforts of co-developers in open source projects. Co-developers are free to participate in projects as they see fit. Almost all co-developers are working for free and their motivation stem from personal interest, personal need, being part of a community, wish for reputation or desire to be respected by a particular peer group. There may be other motivational factors and motivation will be discussed in greater detail at a later stage. It seems that co-developers are motivated by many different things but not by so-called leaders in opens source projects. It might be an exception, when a co-developer is seeking recognition by a particular project leader and does so by following his every wish. There is little evidence to support a theory of direct leadership in open source development, it rather seems that there is evidence of indirect leadership.
Indirect leadership is the maintainer performing his duty of overseeing the development of the project software. The maintainer receives bug-fixes and patches containing new features from co-developers, who are interested in the software. The maintainer serves as a gatekeeper for the project and he single-handedly decides which patches and bug-fixes are accepted into the software. This position gives the maintainer a high degree of control of the direction of development. This is the indirect leadership that a maintainer may exert – a choice between suggested improvements to the project software.

3.4 - Nicole Engard “Practical Open Source Software for Libraries”:

Practical Open Source Software for Libraries by Nicole Engard provides an excellent background of open source movement and its applications for libraries. It is non-technical, assumes no prior knowledge of open source and other definitions of the jargon and descriptions of programs. After the initial introduction to open source and why libraries should get behind it, the book is organized into chapters about various types of open source products.

In the early chapters book, (Engard 2010) addresses popular open source myths, such as fears about security, time commitment, or inferior quality. Another chapter discusses how libraries benefit from open source products. Besides growing budget concerns that may move some libraries toward FOSS, Engard also tries to convey the many similarities between libraries and open source software in mission and history.

After this preliminary chapters introducing open source in libraries, Engard moves onto specific products; this section takes up the bulk of the book. There are applications for operating systems, statistic gathering, instant messaging, screencasting, content management and more.
3.5 - Aditya Tripathi “Open Source Library Solutions”:

Open Source Softwares (OSSs) have come up to liberate Library community from the paws of costly and proprietary softwares. Libraries can implement OSSs in the areas of Library Management Systems, Digital Libraries, E-publishing, Federated Searching, Consortium Management etc.

The book (Tripathi 2000) presents real time experiences of librarians with OSS and demonstrates the capabilities of librarians required on various categories of OSS tools like, Digital Library systems, Library Management Systems, Content Management Systems, E-learning Systems etc.

3.6 - Casey Bisson “Open Source Software for Libraries”:

In May/June 2007, issue of Library Technology Reports Casey Bisson, with the help of Jessamyn West and Ryan Eby, reports on open-source software (OSS) and its use and importance in libraries. Bisson engagingly narrates the history of open source, explains how the OSS “movement” came about, details key players in OSS development and discusses why and how open source can work for libraries.