Open Source Software Peer Review Practices: A Case Study of the Apache Server

Authors: Peter C. Rigby, Daniel M. German and Margaret-Anne Storey

Presented By: Abhishek Dasgupta

What the Paper mainly discusses?

The paper mainly discusses about

1. Two peer review techniques, i.e RTC (Review Then Commit and CTR (Commit Then Review)

2. A comparison between the two processes with archival data.

3. Possible implementation of the processes in real life industrial practice.

What is Review Then Commit?

RTC is a peer review process more used in OSS community. It is basically a technique where a review is made before committing it to the original code. It is actually a process where small chunks of codes are tested before implementation.

What is Commit Then Review?

CTR is just the reverse process of RTC. It is the process where the change is committed before. It is only after the implementation that the review is done. Sometimes this process may become ‘Commit then Whatever’ !!!

Background and Policy Behind RTC and CTR

Apache Server project started by group of volunteered system administrators to fix the original code base of NCSA server.

Shared control repository since all reviews was based on line. The developers never met face-to-face.(like posting blogs)

Core developers given commit rights. New developers have to take rights or reviews from their mentors before committing.... rise of RTC.

RTC became time consuming and onerous process. These gave rise to CTR. RTC was used when the core members was not confident about the changes or the change was being committed by no-core members.

Research Questions raised in the paper

1. Frequency of the reviews and the relation of the reviews to development activity.

2. How many people participated in the reviews and how many reviews were there for the changes.

3. Size of the artefact under review.

4. Review interval or the calendar time to perform a review.

5. How many reviews were successful in finding defects and rectifying it.

Methodologies Involved and the Data Sources :

Mailing list are the main source of data here since the core developer group never met face to face. Discussions are usually conducted on a mailing list as email thread.

For RTC all contributions that has to be reviewed contained the word ‘PATCH’ in the email subject. So all the replies next are concluded as reviews.

For CTR each commit subject begins with ‘svn (or cvs) commit’ in the email subject. So all the replies to it are taken as reviews.

Only the contributions which received a response has been brought into account here.

A random sample of 100 RTC and 100 CTR was collected from the Apache server project.

Two additional Data sources were, manual examination or manual reading of the review and looking at the commit log to determine who was involved in a review.

Extraction Tools and Techniques

Scripts were created to extract the mailing list and version control into a database. An email script extracted the email headers along with sender, in- reply-to and date headers. The messages are threaded in the database following the reference and in -reply-to header. But many messages did not contain these headers thereby giving rise to broken threads which resulted in artificially large number of small threads.

These were reduced by a large extent. For RTC it was reduced from 18.3% to 7.7% and for CTR it was reduced from 15% to 1.8%.

Results From Archival Data

The archival data records shows some results for the questions raised in the paper. Since the data are not normally distributed, its meaningful using the median data.

Frequency : Per month frequency recorded for RTC, RTCA (Review Then Commit Accepted) and CTR are 14, 18 and 19.5 respectively.

Size : Size of the artefact is generally a common measure in inspection. Here the measurement is given in terms of changing lines of codes. The median for RTC, RTCA and CTR are 26, 17 and 15 respectively.
RTC has the largest contribution. 80% contributions has less than 106 changed lines.
In RTCA 80% of the contributions has less than 71 changed lines while that in CTR 80% of the contributions has less than 72 changed lines.

Review Interval : Review interval is the calendar time required to perform a review starting from the first posting of the artefact and till the last review is received and all defects have been rectified. For RTC 80% of the review are made in less than 3.5 days and 50% in less than 19 hours and only 8% of the reviews lasted for more than one week.

Results From Archival Data (Contd.)

Review Interval(Contd) : For CTR the time recorded is the time after commitment till the end of the final review. In 50% of the cases the first review take place within 1.5 hours and 80% of the time within 11 hours. The discussion interval have a median value of 7.5 hours and only 5% issues are found after one week have passed.

Defects : For RTC, 47% contained at least one defect, 7% false positives and remaining no defects. Among the reviews which did not have any defects, 83% had source code defect, 62% contained abstract defect and 47% contained both.
For CTR, 61% contained at least one defect, 9% false positives and the rest, no defect. Among the reviews which did not have any defects, 70% had source code, 62% contained abstract defect and 31% contained both.

Participation : Here the review group is defined on a monthly basis, i.e the the number of individuals who responded to a review in a month. For RTC the median is 14 and for CTR it is 15.

Comparison between CTR and RTC

- CTR is faster than RTC
- CTR has a shorter time interval than RTC
- CTR finds more defects than RTC. (Really So ????)
- CTR is used only when the core developers are committing the change or the review
- group is confident about the change that is going to be made.
- RTC applied to new peoples in the group or people outside the core development group.

Theory of Apache Peer Review :

The paper reported mainly on

1. Review process, policies and structure
2. Reviews finding defects
3. level of participation
4. frequency of reviews and development activities
5. time taken for the review
6. size of the artefact

Based on the above points a theory is presented

Early, frequent reviews (1) of small, independent, complete contributions (2) conducted asynchronously by a potentially large, but actually small, group of self-selected experts (3) leads to an efficient and effective peer review technique(4)

Conclusions

Critical analysis of the OSS style RTC and CTR peer review techniques based on archival data.

The paper provides a theory and methodology that can be potentially used to study OSS review and in more general terms peer review as a whole.

Views for developers for taking peer review more seriously in industrial scenarios.

The paper presents ways so that the theory can be incorporated in industrial practice by breaking up the codes into smaller chunks.

The paper provided a route towards the achievement of high quality software in a limited span of time.

Related Work and Future Course of Study

A recent case study at Cisco used RTC technique. They slightly modified the technique by assigning specific modules to specific reviewers.

Subversion is also following the Apache type of review technique.

In near future how the peer review processes discussed here can be incorporated in industrial scenarios can be a subject of concern. Finding out a way where these techniques can be combined or can be combined with some formal methods is an obvious question whose answer needs to be found out.

References

1. J Asundi and R Jayant. Patch Review processes in open source software development
communities : A comparative Case study

2. C. Bird, A. Gourley, P. Devanbu, M. Gertz, A. Swaminathan.
Mining email social networks. In Proceedings of the 2006 international workshop
on Mining software repositories, pages 137-143, 2006.

3. J.Cohen. Best Kept Secrets of Peer Code Review. Smart Bear Inc., Austin,
TX, USA, 2006

4. P.C. Rigby and D.M. German. A preliminary examination of code review processes in
open source projects. Technical Report DCS-305-IR, University of Vicoria, January 2006.

Questions