

CONVERGENT CONTEMPORARY SOFTWARE PEER REVIEW PRACTICES

Presented by

Tresa Varghese Rose(101017725)



Carle
UNIVER

Introduction



- Peer review is the practice of inspecting code by other authors to find defects and thus improve software quality.
- Even if bug inspection is the main aim, the traditional review practices has been shown limitation of software adoption and review efficiency.
- Contemporary or modern peer review follows less rigid practices.
- While different companies follow various contemporary light weight review, no study was conducted on the efficiency of each.
- In this paper, a detailed study on contemporary software review with varying domains, organisations and development processes is done.

Research Questions

- How do the parameters of peer review differ in multiple disparate projects?
 - What peer review process (e.g., Fagan inspection vs Commit-then-review) does the project use?
 - How long do reviews take and how often are reviews performed?
 - What is the size of artifact under review?
 - How many people are involved in review?
 - How effective is review in terms of problems discussed?
 - Does review spread knowledge about the system across the development team?



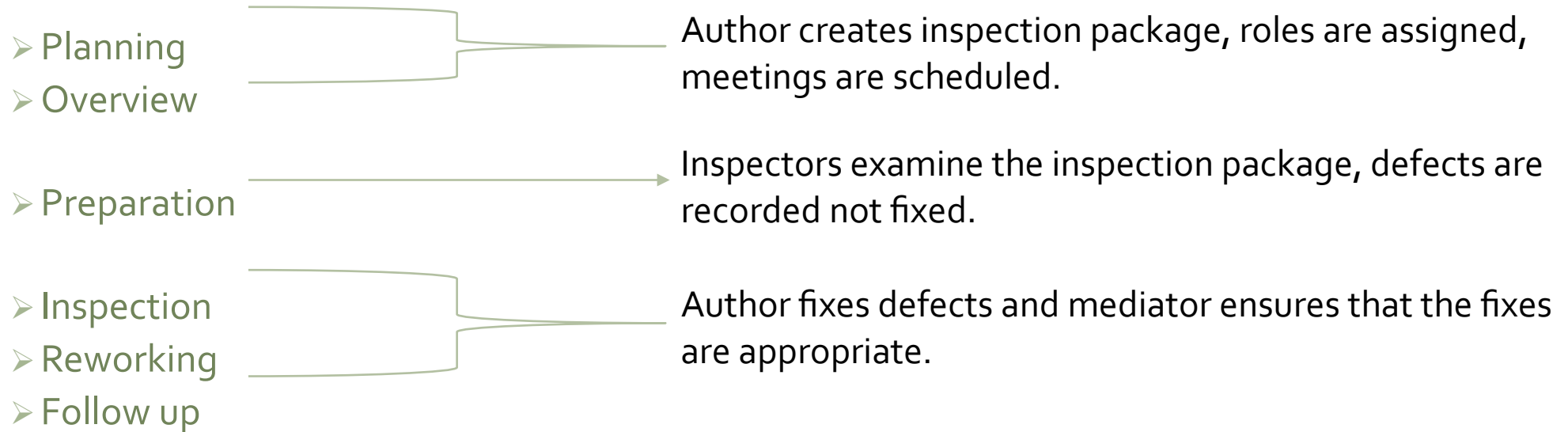
Background

- 3 types of peer reviews are performed :
 - Traditional Software Inspection
 - OSS email based peer review
 - Light weight tool supported review
- Data source:

Traditional Software Inspection	OSS email based peer review	Light weight tool supported review
Lucent project.	Apache	Advanced Micro Devices (AMD)
	Linux	Microsoft (Bing, office, MS sql)
	KDE	Google led Projects (android and chromium OS)

Software Inspection

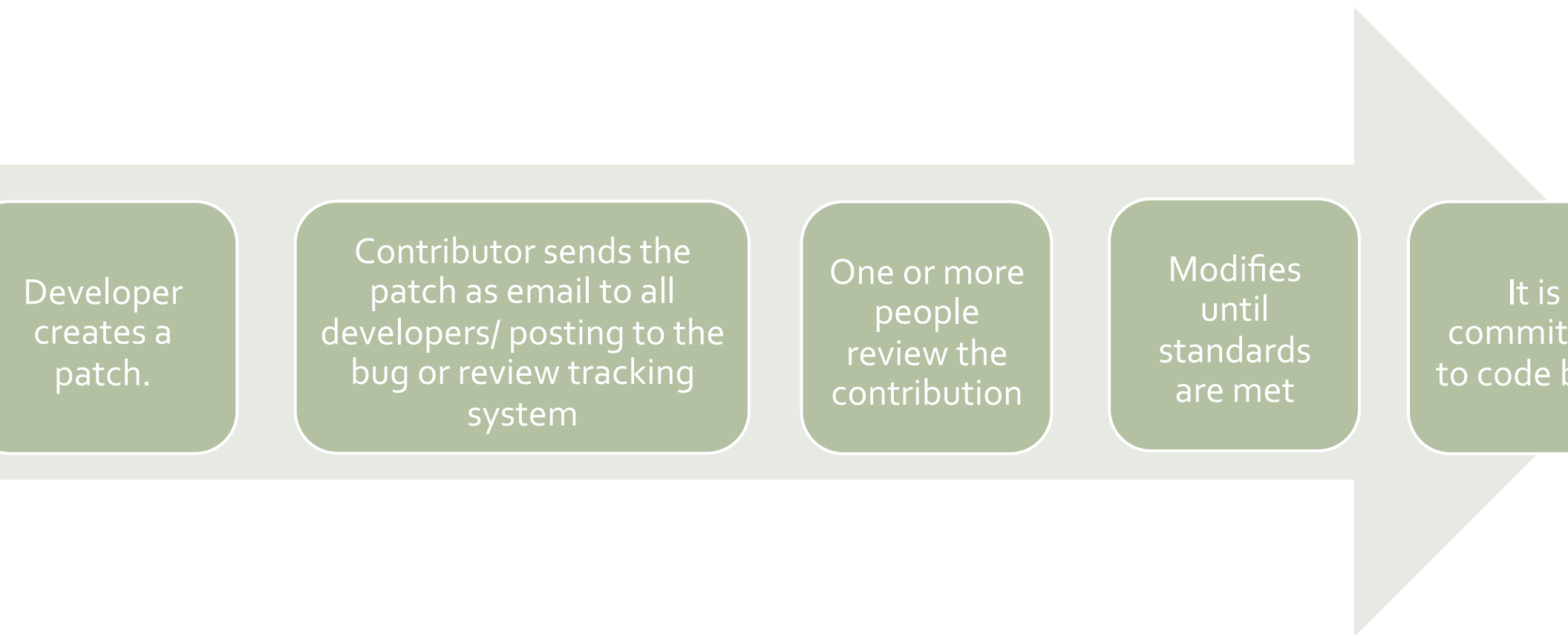
- Formal type of review.
- Steps involved



- Comparison Data – Data collected from inspection experiments at Lucent.

Open Source Software Peer Review

- Peer review is the best quality assurance method followed by OSS projects.



- **2 types** of reviews happen in OSS Peer Review

- RTC (Review Then Commit):

- Contributions will be reviewed and once approved, it will be committed.
- Many contributions are ignored or rejected and never make it to code base.

- CTR (Commit Then Review):

- Trusted developers can commit their contributions. The core developers will then review all commits.

- **Comparison Data**

- Data from 6 OSS project

- Subversion version control system
- Linux OS
- The free BSD OS
- KDE desktop environment
- Gnome desktop environment

Peer Review at Microsoft

- A centralized internal tool, CodeFlow is used for review before submitting the changed code to version control system.

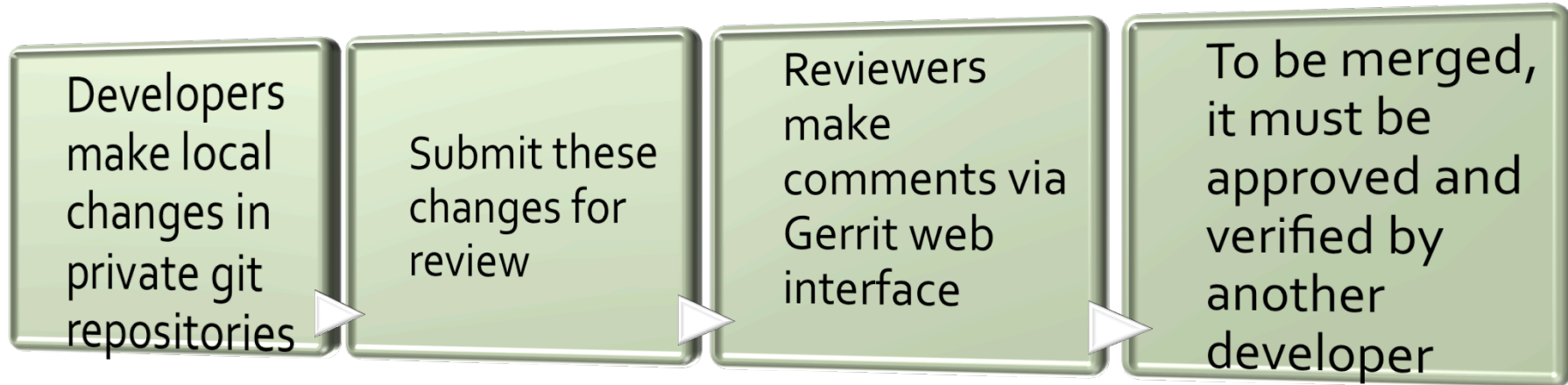


- Data : 3 Projects from Microsoft

➤ Bing , MS Office 2013, MS SQL Server

Google-Based Gerrit Peer Review

- Gerrit is an OSS, code review practice implemented internally in Google.
- Stands between developers private repository and centralised repository.



- States of review process – Verified, Approved, Submitted.
- Data – Results from Google-led OSS projects that use Gerrit tool.

➤ Android, Chromium OS

AMD and Code Collaborator

- Code Collaborator based peer review practice on an internal AMD project.
- Steps Involved:
 1. Author uploads software artifacts for review in the web interface.
 2. Reviewers are assigned to the review.
 3. Review discussion occurs and problems fixed.
 4. Once review is approved, it is committed.
- A review must be approved by 2 reviewers.
- Data :
 - Quantitative results from the use of code collaborator at AMD. But AMD dataset is limited.

Contemporary Peer Review Practices

- Steps involved in contemporary peer review practices:
 - The author creates a change and submits it for review.
 - Developers discuss the change and suggest fixes. The change can be re-submitted multiple times to deal with the suggested changes.
 - One or more reviewers approve the change and it is added to the “main” version control repository. The change may also be rejected.

Methodology and Data

- Yin's multiple cases study methodology is used.
- Analytical generalizations can be made from the findings.(helps researchers to develop a theory or framework of findings)

Data Extraction

- Lucent – data from reviewers on a compiler project is collected by attending meetings and used as comparison data.
- OSS project – review data was extracted form mailing list and considered valid if there is a change and 1 or more emails from reviewer.
- Microsoft data was extracted from code flow tool which stores all data regarding code reviews.
- Google chrome and android - Gathered information abt authors and reviewer's activity, files changed, comments made and dates of submission and completion from gerrit server.
- AMD dataset does not have all parameters needed, so review discussions with at least 1 reviewer is considered.

Multiple Case Study Findings

- **Convergent Practice 1:** Contemporary peer review follows a lightweight, flexible process
 - Some OSS projects and all the software firms we examine use a review tool, which makes the process traceable.
 - Contemporary reviews are typically conducted asynchronously and measures of review are recorded automatically.
- **Convergent Practice 2:** Reviews happen early (before a change is committed), quickly, and frequently.
 - For most of the projects studied review interval ranges from few hours to a day.

Project	Review Interval (Average)
Lucent	10 days
Apache & Linux	Few hours to a day
AMD	17.5 hours
Microsoft Bing	14.7 hours
SQL	19.8 hours
Office	18.9 hours
Google Chrome	15.7 hours
Android	20.8 hours

- No of reviews per month is very high compared to traditional inspection practices but varies with the stage, development style, and size of the project.
- While review frequency in AMD & Bing is increasing by each month, Android & Office shows fluctuations. But Chrome & SQL shows a relatively stable no of reviews.

Convergent Practice 3: Change sizes are small.

➤ Shorter intervals can be achieved if the changes made are small.

Projects	No of Lines Changed
OSS	11 to 32
Android and AMD	44
Apache	25
Linux	32
Lucent	263
Chrome	78

Convergent Practice 4: Two reviewers find an optimal number of defects.

➤ A minimal increase in the number of comments about the change.

- **Convergent Practice 5**: Review has changed from a defect finding activity to a group problem solving activity.
 - On OSS projects, the discovery of the defect is not the focal point. Instead developers discuss potential defects and solutions to these defects.
 - Contemporary practices found similar to OSS : number of defects are not explicitly recorded as it can distract developers from primary task of fixing the defects found in review.
 - 3 alternative measures suggested for review effectiveness
 - The number of comments during review is an upper bound on the number of defects found per review.
 - A better estimate is the number of comment threads.
 - A lower bound on the number of defects found in a review is the number of artifact resubmissions

Benefits & Limitations

- **Sharing Knowledge** through review across the development team.
- A preliminary **measurement of knowledge** spreading is done by checking no of files a developer has modified, the number of files a developer has reviewed and the total number of files he knows about.
- **The number of defects found** during review is known to be a limited measure of review effectiveness.
- Since study was done on diverse projects using different processes, tools etc **the dataset will not be same as the one in an experimental setting.**
- Some of the reviews in Microsoft projects were **not given proper attention** from developers.
- AMD and Lucent datasets were not raw ones but just a **summary.**

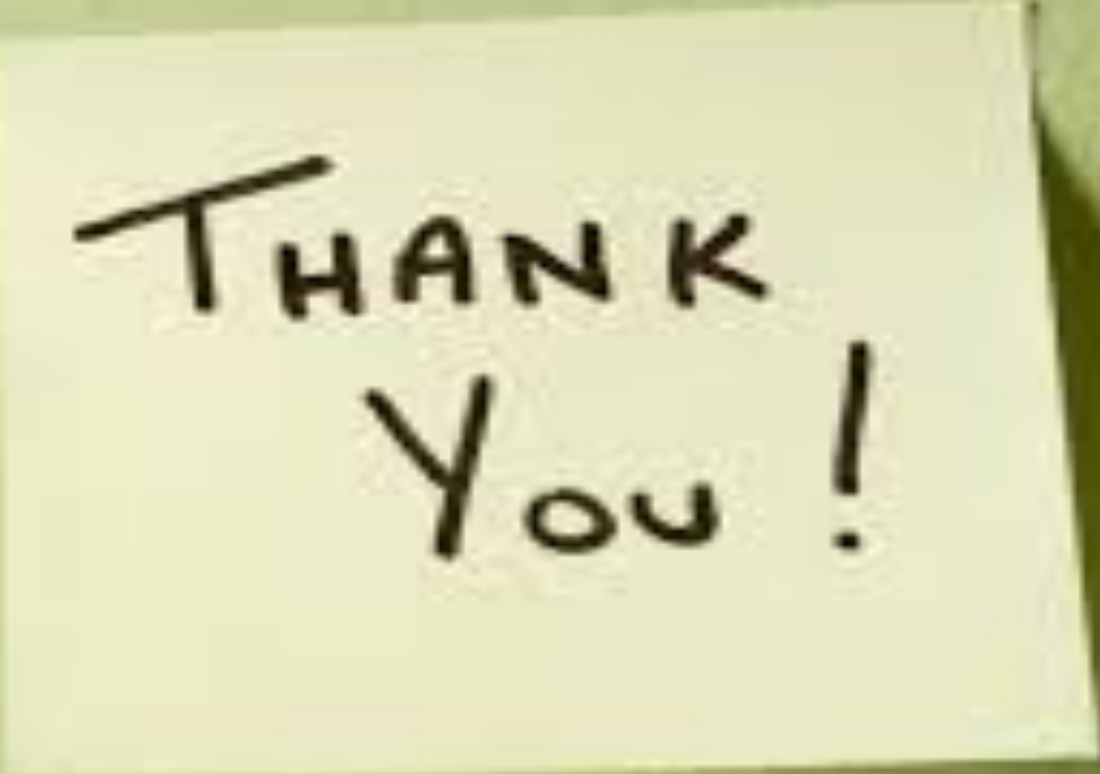
Conclusion

- Contemporary software inspections are more lightweight review compared to the formal inspections.
- They are performed regularly and quickly just before code is committed.
- Contemporary review usually involves 2 reviewers but can vary depending on the complexity of change.
- Contemporary reviewers prefers discussion and fixing code over reporting defects.
- Tool supported review provides the benefits of traceability, when compared to email based review, and can record implicit measures, when compared to traditional inspection.
- The degree of knowledge spread among the developers.

Discussion



- Do you think contemporary peer review is more effective than traditional inspections?
- Is group problem solving activity more important than defect finding?
- What can be some of the future works?



THANK
You!