

Deriv

Peer Reviews - Process Description

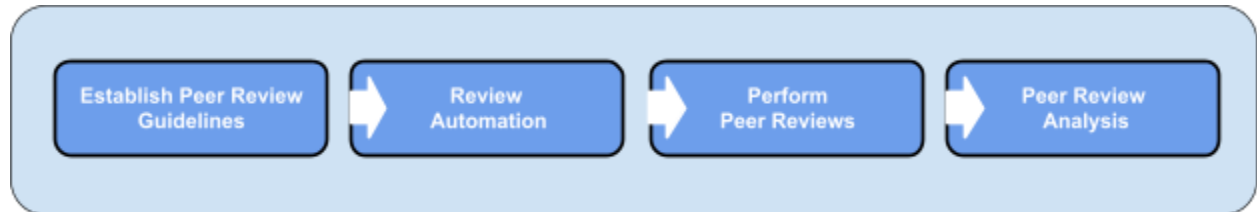
CONTENTS

1. VERSION CONTROL	2
2. OVERVIEW	3
3. ESTABLISH PEER REVIEW GUIDELINES	4
3.1.1 OVERVIEW	4
3.1.2 TEAMS	4
3.1.3 PROCESS DESCRIPTION	5
3.1.4 INPUTS	5
3.1.5 PURPOSE, KEY ELEMENTS AND OUTPUT	5
3.2 REVIEW AUTOMATION	6
3.2.1 OVERVIEW	6
3.2.2 TEAMS	6
3.2.3 PROCESS DESCRIPTION	7
3.2.4 INPUTS	7
3.2.5 PURPOSE, KEY ELEMENTS AND OUTPUT	8
3.3 PERFORM PEER REVIEWS	9
3.3.1 OVERVIEW	9
3.3.2 TEAMS	9
3.3.3 PROCESS DESCRIPTION	9
3.3.4 INPUTS	10
3.3.5 PURPOSE, KEY ELEMENTS AND OUTPUT	11
3.4 PEER REVIEW ANALYSIS	12
3.4.1 OVERVIEW	12
3.4.2 TEAMS	12
3.4.3 PROCESS DESCRIPTION	13
3.4.4 INPUTS	13
3.4.5 PURPOSE, KEY ELEMENTS AND OUTPUT	14
4. RACI MATRIX	15
5. REFERENCE LINKS	16

1. VERSION CONTROL

Version No.	Date of Last Change	Preparer	Approver
1.1	24/10/2021	Prince Coching Mohammadreza Ghorbani	

2. OVERVIEW

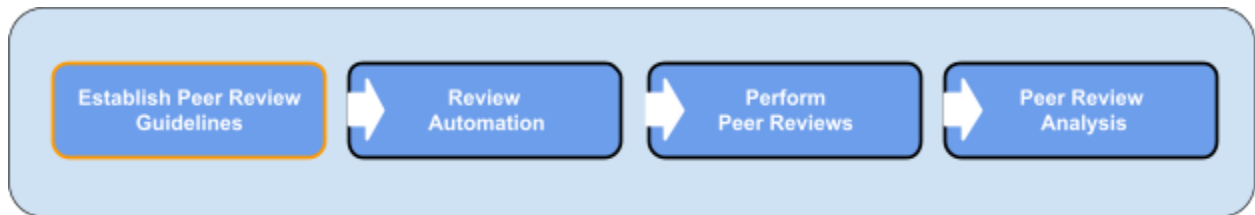


Peer review acts as a process gateway in the delivery of work products. It is intended to assert the quality of the work product but is not the same as a Quality Assurance step in the process, Peer review as is hinted in the name is performed by peers of the work product producer. Having peers review work has the benefit of not only helping ensure quality but also helping to share knowledge between team members. Peer review generally occurs in the Execution step of a project but is not limited to this.

3. ESTABLISH PEER REVIEW GUIDELINES

3.1.1 OVERVIEW

Peer reviews are conducted in conjunction with some standard guidelines. These guidelines may vary on different teams. A guidelines document is required to be present in each teams that perform peer reviews



3.1.2 TEAMS

The following teams will be involved at this stage:

- Design Team
- Devops Team
- FrontEnd Team
- BackEnd Team
- Mobile Development Team
- QA Team

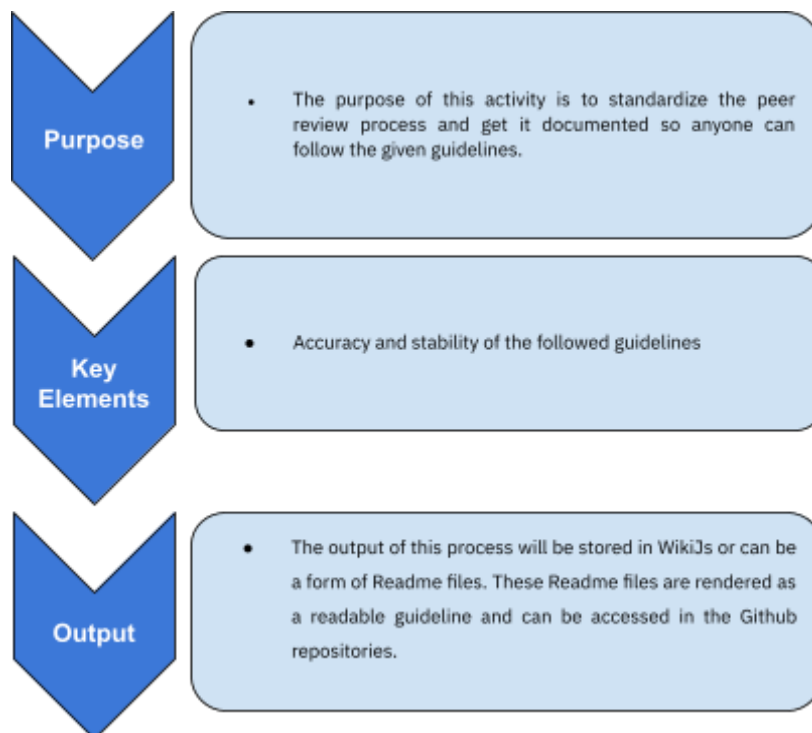
3.1.3 PROCESS DESCRIPTION

- Documented procedures are stored in WikiJs for most of the teams
- FrontEnd, BackEnd, Mobile Development and QA team have extra documentation that lies within specific platforms. These documents are usually Readme files that contain standard coding patterns and components structure of each platform's code base.

3.1.4 INPUTS

- Best practices guidelines in coding.
- Third party plugins documentation.
- Design patterns and formats
- User Journey in Critical Path.

3.1.5 PURPOSE, KEY ELEMENTS AND OUTPUT



3.2 REVIEW AUTOMATION

3.2.1 OVERVIEW

Creating an automated review can make the lives of our team members easier, not only because it does the common checks for us but it also eliminates the time to manually check coding errors including misspelled variables, missing dependencies and incorrect formats.

The Tech team (Front End, Backend, Mobile Development and QA) will have their automated checks on Github powered by CircleCI. However, the Frontend team will have an extra layer of automated checks to ensure code clarity and to strengthen security vulnerabilities. SonarCloud will come into play to handle these automated checks.



3.2.2 TEAMS

The following teams will be involved at this stage:

- FrontEnd Team
- BackEnd Team
- Mobile Development Team
- Designers
- DevOps
- QA

3.2.3 PROCESS DESCRIPTION

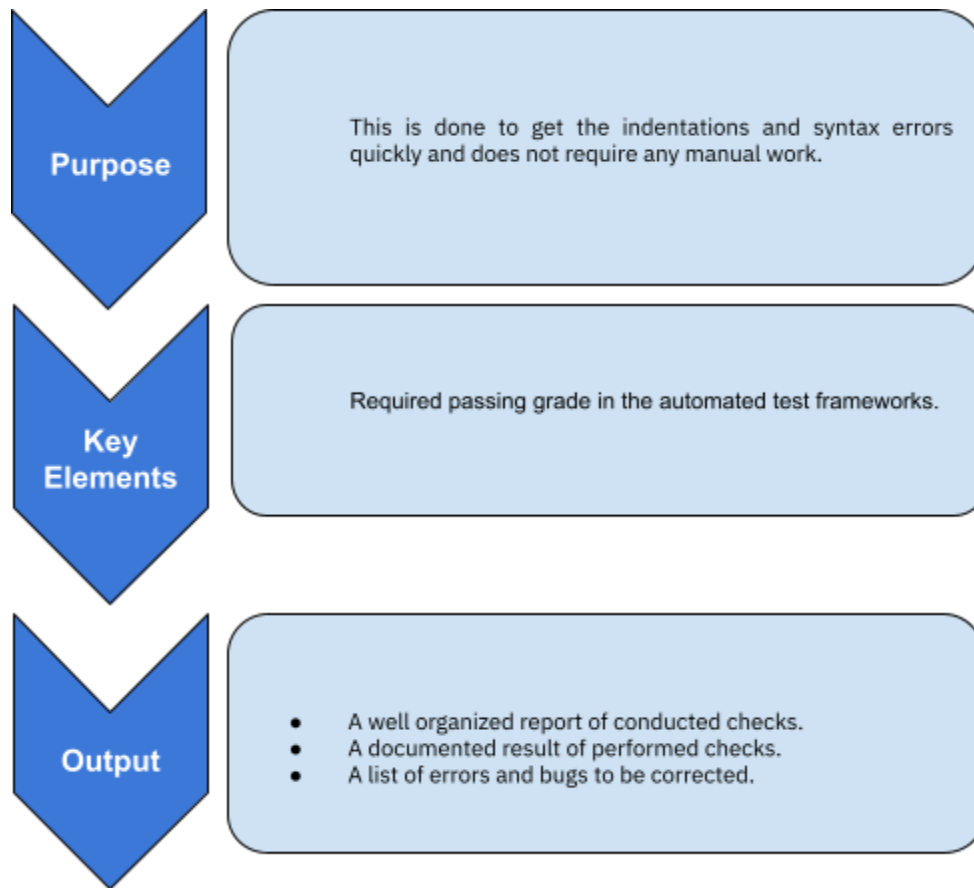
- Developers or QA Engineers should prepare the platform requirements and gather all the necessary best coding practices guidelines.
- An assigned developer to set up the automated checks should create a configuration file to handle the check automation.
- DevOps should ensure that the configuration file is properly set up and working as per the specifications.

3.2.4 INPUTS

Inputs for setting up the automated review

- System requirements
- Automated test configurations
- Output directory for reports.

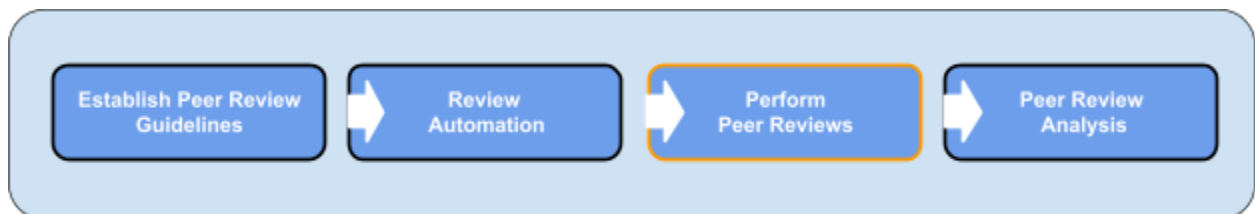
3.2.5 PURPOSE, KEY ELEMENTS AND OUTPUT



3.3 PERFORM PEER REVIEWS

3.3.1 OVERVIEW

The actual process of performing a peer review is manual. Each team has a divided field for each platform and each platform has its own reviewers. Code reviewers will thoroughly check the code or design based on the guidelines. These reviews will take a higher consideration on what existing features of a current solution have and how the new solution affects it.



3.3.2 TEAMS

The following teams will be involved at this stage:

- FrontEnd Team
- BackEnd Team
- Mobile Development Team
- Design Team
- QA Team
- Compliance
- CopyWriters

3.3.3 PROCESS DESCRIPTION

Each team should conduct their peer review in their own process but should update the status of the review in Redmine cards. These cards won't go to the next process until 1-2 votes are present in the card.

FrontEnd, BackEnd and Mobile Development teams

- Developers should check the readiness of the pull request of their peers
- Pull requests that have code conflicts or CI build failures should go back to the respective coders and start working on the fix.
- Reviewers should follow the guidelines for coding best practices and point out issues or change recommendations in a form of comment using GitHub Review features.
- Slack messages should be sent to the developers to inform them about the review results. (Sometimes automated emails are not enough)

Quality Assurance (QA) Team

- Reviewers of the redmine cards should be aware of the contents of the tasks like the design and flow of the feature.
- Reviewers should check the test performed whether there are checks missing or additional checks are needed.
- If there are missing test procedures and/or any additional checks, then the reviewers should add the items to the checklist to redo the testing process.

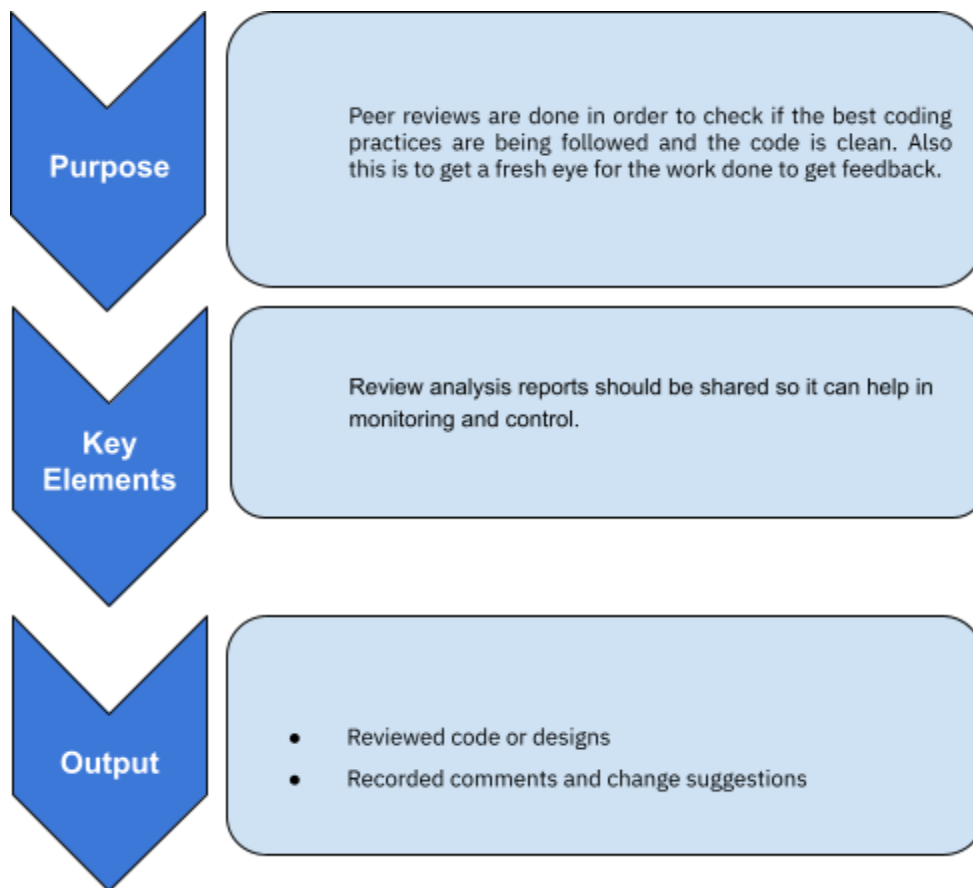
Design Team

- Reviewers will check the Figma file containing the layout and designs based on specifications.
- Reviewers will add comments in the Figma file for update requests.

3.3.4 INPUTS

- Redmine cards with test links and configurations.
- GitHub Pull Requests
- Design Mockups
- External widgets & APIs

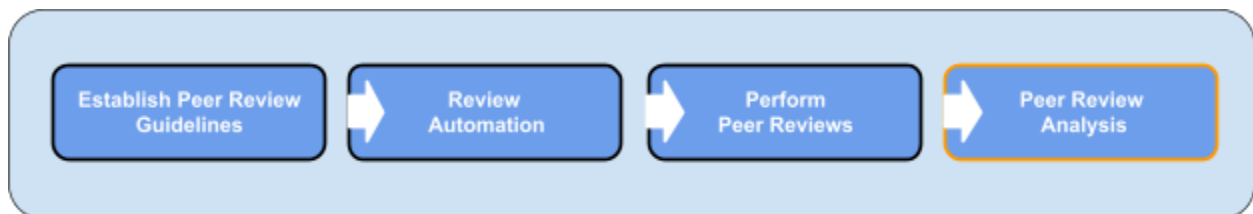
3.3.5 PURPOSE, KEY ELEMENTS AND OUTPUT



3.4 PEER REVIEW ANALYSIS

3.4.1 OVERVIEW

It is important for us to monitor the peer review process. Once we have good metrics in place we can determine if certain review gateways are worth the effort and how much time the peer review process is taking. Monitoring of peer review output for trends in identified defects will allow the implementation of preventative measures.



3.4.2 TEAMS

The following teams will be involved at this stage:

- FrontEnd Team
- BackEnd Team
- Mobile Development Team
- Design Team
- QA Team

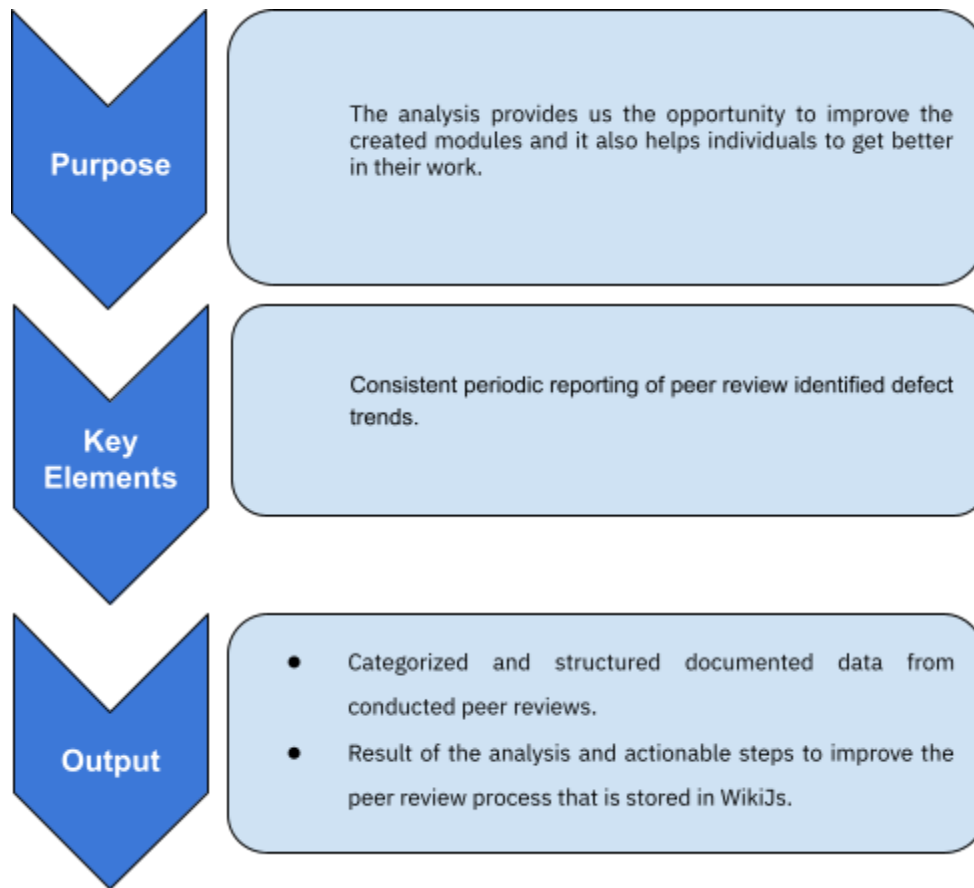
3.4.3 PROCESS DESCRIPTION

- Gathering data related to conducted peer reviews such as changes, comments and suggestions.
- Data gathering can be automated or manual depending on the team. Tech team should gather data from automated records in GitHub Pull Request Reviews. While other teams like design and QA might need to manually gather their data from Redmine and Figma.
- Gathered data will be categorized based on measuring points of the automated and manual peer reviews.
- The analysis should be conducted every quarter and the reports should be documented and stored in WikiJs.

3.4.4 INPUTS

- Comments and suggestions left by the reviewer, especially in GitHub pull requests.
- Checklist of missing items and missed out processes.

3.4.5 PURPOSE, KEY ELEMENTS AND OUTPUT



4. RACI MATRIX

Process Steps	Establish Peer Review Guidelines	Review Automation	Perform Peer Reviews	Peer Review Analysis
Product Design	R	R	R	R
Backend	R	R	R	R
Web/ Mobile Frontend	R	R	R	R
DevOps	R	R	R	R

(A=Accountable , R=Responsible , C=Consulted , I=Informed)

5. REFERENCE LINKS

Document	Links
How to do Code Reviews like a human	https://mtlynch.io/human-code-reviews-1/