Formal Technical Review Process Enhancement

Eu Felix, Peh Wei Wooi, Ooi Mei Chen, Liu Keping

felix.eu@intel.com, wei.wooi.peh@intel.com, mei.chen.ooi@intel.com, keping.liu@intel.com

Abstract

The Formal Technical Review (FTR) is a software quality control activity with the purpose of ensuring the software fulfills specified requirements and predefined standards. In the past, many organizations have deprioritized the importance of reviewing feedback because they did not expect it to show a significant value added to software quality or bring any tremendous benefits to the organization. Despite the proven data that reviews significantly increase software release quality [Ref 1], they may be viewed as a redundant part of the software release process.

Based on years of experience in the quality assurance industry, an efficient FTR process has consistently proven its ability to improve the software quality in various aspects such as business requirements, architecture design documents, technical requirements, software release documents, test cases procedures, etc. The quantitative data in terms of cost-saving metrics and defects will be further explained in section 5. In a nutshell, the formal technical review is a key factor in meeting the quality release criteria and reducing the unnecessary cost (to fix the mistakes) incurred in the project, hence an effective FTR process and tool should be implemented to ensure the success of the project which leads to the success of the organization.

Biography

Eu Felix is a Software Quality Engineer at Intel Corporation based in Penang, Malaysia. He has held the Lean Six Sigma Green Badge since 2019, is a certified Software Quality Engineer (CSQE) from ASQ and holds a Degree in Computer Science from the University of Bolton in the UK.

Peh Wei Wooi is a Platform Validation Lead at Intel Corporation based in Penang, Malaysia. He is certified as the ISTQB tester and holds a Degree in Information Science from UKM, Malaysia.

Ooi Mei Chen is a Software Quality Engineer at Intel Corporation based in Penang, Malaysia. She holds a Degree in Computer Science from Universiti Tunku Abdul Rahman.

Liu Keping is a Technical Leader in Software Quality Assurance at intel Corporation based in Shanghai, China. She is a certified CMMI assessor, ISO internal assessor, ASPICE internal assessor, and CSQE, and gained 6 Sigma Orange Belt and CPMP certification in 2009. She holds a master's degree in Computer Science and Technology from Central South University in China.

1 Introduction

In this introduction, we discuss what Formal Technical Review (FTR) is and the importance of having it in the entire software development lifecycle by defining the objectives. We briefly go through the high-level overview of the different types of reviews currently available and move on to provide detail of the phases available in the FTR process.

In Section 2, we discuss the roles and responsibilities of each role that will be involved in the formal technical review. It is important to understand how each role plays its part and contributes to the success of the review. Section 3 describes the overall methodology for conducting an effective formal technical review, for example, setting up the meeting, record keeping, review reporting, review guidelines, and the recommended formal technical review tools which summarize the "best-known method" for the formal technical review process.

Section 4 will recommend several enhancements which we believe are useful based on past user experience, and how they can benefit organizations upon implementation. The two process enhancements are FTR Guidelines and FTR Tools. FTR Guidelines focus on how to reduce waste by implementing lean practices and avoiding process and roles duplication, while the introduction of the FTR tool brings more value by having a centralized repository for all the review comments, data consolidation, email notification, versioning & tagging, FTR dashboard and so on that enlightens exceptional and outstanding user experience. These recommendations are very important for an organization as they not only help to improve the FTR review efficiency (via processes) but also improve the velocity with the help of the FTR tool. In section 5, we provide some examples of the review metrics from various aspects which include user experience, establishing organizational benchmarking, and defining the quality data. Finally, in Section 6, we summarize and provide directions for future work and areas to research.

1.1 What is the Formal Technical Review (FTR)?

The Formal Technical Review (FTR) is a software quality control activity to ensure that the software meets the specified requirements and predefined standards. In the past, many organizations have deprioritized the importance of review feedback because they didn't envision it to show a significant value added to the software quality or bring any tremendous benefits to the organization. Despite the proven data that reviews level up software release quality, they may be viewed as a redundant part of the software release process.

Based on years of experience in the quality assurance industry, an efficient FTR process has consistently proven its ability to improve software quality in various aspects such as business requirements, architecture design documents, technical requirements, software release documents, test case procedures, etc. For instance, during the earlier stage of business requirement review, the customer and respective stakeholders will be engaged in the review and all the misconceptions or misalignment can be fixed earlier before flowing into the next stage which is technical requirement creation. In the nutshell, the formal technical review is a key factor in gauging the quality release criteria and reducing the unnecessary cost (to fix the mistakes) incurred in the project, hence an effective FTR process and tool should be implemented to ensure the success of the project which leads to the success of the organization.

1.2 Objectives

The high-level objectives for carrying out the formal technical review process are:

- To discover defects in function, logic, or implementation of the software.
- To ensure the software implementation meets the requirements and predefined standards.
- To implement and archive software in a uniform manner.
- To make projects more traceable such as the feedback and verification of changes are stored in a consolidated repository.

1.3 Types of Reviews

There are two types of reviews: Formal reviews and Informal reviews. Both types of reviews can be useful depending on the available resources.

Formal Reviews	Informal Reviews
Larger Group	Smaller Group
The review is scheduled beforehand. Sufficient time is given to the team members for preparation	The meeting is commonly scheduled at team members' convenience
Follow the process with specific formal agenda	Conducted per the need of the team with informal agenda
The review consists of a professional team that identifies and corrects errors in the software model	Equivalent to a buddy check. The main purpose is to detect defects, generate ideas/solutions, or quickly resolve minor problems
The review should not exceed two hours	This review is in between 1-2 hours

Table-1 Types of Reviews

1.3.1 Different Types of Formal Reviews

The table below explains three different types of formal technical reviews which can be applied throughout the Software Development Lifecycle. Table-2 shows the overview of the different types of formal reviews currently available and their respective goals.

Type	Definition	Goals
Walkthrough	Led by the author to understand and collect comments. The content of the document is explained step by step by the author, to accomplish agreement on changes or to gather information. The participants are chosen from different departments and backgrounds. If the audience represents a broad section of skills and disciplines, it can give a guarantee that no crucial defects are "missed". It is helpful for higher-level documents, such as requirements specifications and architectural documents.	• A preliminary review of the work product. Either first draft or clean code compiles. • To present the document to stakeholders both within and outside the software discipline, to evaluate conformance to a standard, and gather information. • To describe and gauge the contents of the document. • To initiate a basic understanding of the document. • To inspect and discuss the validity of suggested solutions and feasible alternatives.
Technical Review	The review focuses on the technical content of a document and is led by a trained moderator or technical expert. The review is intended to identify discrepancies in specifications and standards and focus on defect	• Determine the suitability of the work product for its intended use. • To assess the benefit of technical concepts and alternatives in the product and project environment.

	identification based on referenced documents. Participants can be technically qualified personnel such as architects, chief designers, and key users. This is openly performed as a peer review without management participation.	• To set uniformity in the use and representation of technical concepts. • To gain consensus and build confidence in the product. • To ensure at an early stage, that technical concept are used precisely. • To notify participants of the technical content of the document.
Inspection	The most formal review type and led by a trained moderator to identify issues in a work product. The document under inspection is prepared and checked thoroughly by the reviewers before meeting by using the rules and checklists. In the inspection meeting, the defects found are captured.	• Like Review but beyond that search for anomalies • To help the author to enhance the quality of the document by detecting potential defects. • To remove defects efficiently and rapidly. • To improve product quality by producing documents with a higher level of quality. • To create a common understanding by exchanging information among the inspection participants.

Table-2 Different Types of Formal Reviews

1.4 Different Phases of Formal Technical Review

An FTR process generally breaks down and is implemented into different phases to ensure process integrity. Figure-1 below shows the FTR principally takes place in the well-thought-through approach that includes 6 different steps that are essential to assure and inspect software quality, efficiency, and effectiveness.

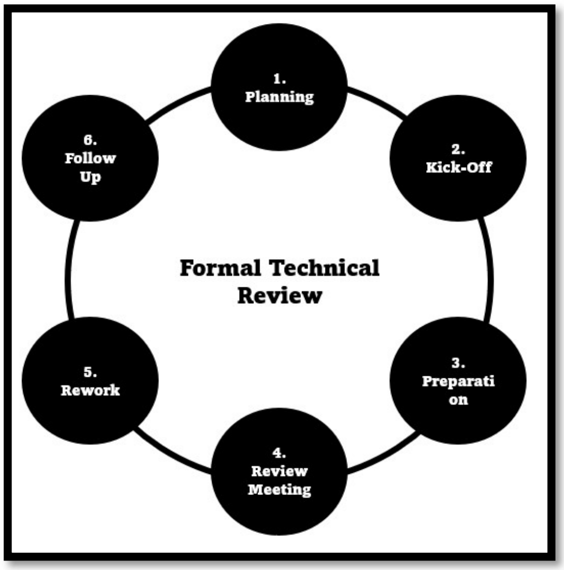


Figure-1 Formal Technical Review Phases

Phase	Descriptions

Planning	Commonly starts with a ‘request for review’ by the author to a moderator. The moderator also performs entry checks and exit criteria.
Kick-Off	The main purpose is to get all participants in sync and understand the intention of this kick-off. The entry result and exit criteria are talked through as well in this meeting. This phase brings a greater level of understanding of the team about relationships between documents under review and other documents. During the meeting, all documents under view, source documents, etc. will be distributed.
Preparation	The participants (reviewers) will work solitary on the document under review. They mainly identify or inspect for any defect/error, and offer their opinion, which is later logged or captured. Spelling or cosmetic errors are also tracked on the document under review but not mentioned during the meeting.
Review Meeting	This phase commonly involves three different activities: logging, discussion, and decision.
Rework	The author primarily revises the document that is under review based on the defects that are detected and improvements being proposed in the review meeting. Changes that are made to the document must be easy to identify during the follow-up phase, consequently, the author is required to specify what changes are made.
Follow-Up	Moderator to ensure all satisfactory actions need to be taken on all logged defects, improvement suggestions, and change requests. Ensure the exit criteria are met.

Table-3 Details of the different phases in Formal Technical Review

2 Roles and Responsibilities

The finest formal technical review commonly comes from well-structured teams being led by moderators or review leaders that are professionally trained. For the review process, there are several roles and responsibilities being defined and required within the review meeting. Table-4 describes the discrete roles and responsibilities for each profession that will be involved in the formal technical review. It is important to understand how each of the professions plays their roles and its contribution to the success of the review.

Role	Responsibilities
Author	An individual or a team who has written the “document under review”. The author needs to • Ensures the readiness of work product for review. • Assists moderator in planning for review. • Prepares for briefing in the meeting. • Fixes all defects identified and resolves all issues raised during the meeting review. • Identify the right subject method experts for the work product review.
Moderator	Known as review leader. Organizes and leads the review process. • Work closely with the author to schedule the meeting review. • Coordinates with the author and checks the entry and exit criteria for review.

	• Ensures the review discussion is focused on the work product and not the author. • Briefs inspection team roles. • Distributes inspection work product to reviewers. The individual needs to be trained to become a certified moderator. The moderator training course can be either enrolled in-house or public. In-house: The certification is only recognized within the organization. The training is customized by the organization to fit its own needs only. Public: The certification is recognized by one or more organizations. The training is commonly used by most organizations.
Reviewer	Scrutinize the document in accordance with the business specifications, standards, and domain knowledge. Verifying the work product's completeness, and correctness, also make suggestions.
Reader	A person who reads the work product in the review meetings.
Recorder/Scribe	A person who records the defect and suggestions/feedback during the review meetings after consensus is reached. Read back recorded defects to ensure the information logged is clear, complete, and correct.

Table-4 Roles and Responsibilities in FTR Process

3 Conduct an Effective Formal Technical Review

An effective FTR ensures the improvement of the software quality by detecting the defects upfront (either due to miscommunications or human errors) to reduce the risk of extra efforts and costs that might potentially be incurred to the project.

3.1 The Review Meeting

- Every review meeting should be conducted by considering the constraints below:
    - Involvement of people in the review.
    - Sufficient advance preparation is allocated (Approximately 2 hours of work per person). If the reviewers do not have adequate preparation prior to the review date, the recommendation is to postpone the review meeting.
    - Review meeting duration should be 2 hours or less.
- Review focuses on a small group of modules rather than the entire design.
- Review focuses on the work product, not the author/designer.
- At the end of the review meeting, all participants must:
    - Accept the work product without any modification.
    - Reject the work product due to crucial error (Requires another round of review).
    - Accept the work product provisionally (Minor errors require correction, but no additional review is needed).

NOTE: The moderator plays an important role in driving the participants and closing all the misaligned opinions. This is to ensure that all participants have the same consensus on the above decisions. Conducting approval voting (The highest votes speak for the majority decision) is one of the common techniques to be used during the review meeting when having a disagreement.

3.2 Review Reporting and Record-Keeping

- During the review meeting, the recorder actively captures all issues and feedback that have been raised.
- All these issues and feedback are consolidated in a review list after the review meeting.
- A summary of the review findings list for formal technical review now is ready to be shared.

3.3 Review Guidelines

A clear and well-defined review guideline should be established and distributed to all participants in advance (1-2 weeks) before the formal technical review. The expectation of participant behavior and process for review should be clearly stated that all the participants agreed upon and then followed. This is to set accurate expectations for the review process during the review.

Roles	Responsibilities
Author	• Limit the number of participants and insist upon advance preparation ❖ Only the related participants are invited. • Create a checklist ❖ The checklist is to ensure the review meeting is conducted more structurally and only focuses on critical issues. • Assign resources and time ❖ Ensure the formal technical review is scheduled as a must-have task in the software development lifecycle.
Reviewer	• Review the work product, not the manufacturer/producer ❖ Defects should be pointed out constructively. ❖ The attitude of the meeting should be loose and beneficial. ❖ Avoid finger-pointing, insulting, or belittling.
Moderator	• Set the agenda accurately ❖ Try to keep the review on track and on schedule. Control the timing if the meeting starts drifting. • Don't attempt to solve every problem ❖ A review is not a problem-solving session. Separate the problem areas, acknowledge the problem, and solve the problem after the reviewing meeting. • Restrain argument and rebuttal ❖ When an issue is raised by a reviewer, there may not be general agreement on its impact. Record down the issue for offline discussion instead of wasting time debating the question. ❖ Involve relevant parties to follow up on the issue discussion after the meeting. If the involved parties can't come to an agreement, assign it to the Program Manager to make the final decision.
Recorder/Scribe	• Write the notes ❖ It is recommended for the recorder to write down the notes on a (physical or virtual) whiteboard so that wording and priorities can be assessed by other reviewers as information is recorded.
Program Manager	• Provide relevant training to all reviewers ❖ To ensure the reviewing meeting is effective, formal training should be provided to the participants

Table-5 Review Guidelines for Each Role

4 Recommendations for FTR Process Enhancement

There is no doubt that implementing the FTR process has helped the organization to gain throughout the Software Development Lifecycle (SDLC). The FTR process is not something that is carved in stone and different people will have different interpretations and expectations. The FTR process enhancement discussed below is based on actual work experience which we believe will further assist the organizations to streamline their resources, time, and budget in software project management. We focus the process enhancement on two specific areas, i.e., FTR Guidelines and FTR Tool.

4.1 FTR Guidelines

4.1.1 Review Checklist

- Create separate templates for review checklists for documentation and source code reviews
- It is recommended to make the review checklist a mandatory activity for each review.
- Reviewer can fill up the checklist to ensure nothing is forgotten in the review process.

4.1.2 Merging Roles and Responsibilities

- To keep the review meeting participants as lean as possible, it is advisable to combine some of the FTR roles and responsibilities with the premise of no conflict of interest.
- The moderator, reader, and recorder roles can be handled by one person.
- The author and moderator should not be handled by the same person.

4.1.3 Avoid Misjudged and Achieve Uniformly Consensus

- The author is not allowed to close the defects and comments by themself.
- The defects and comments must be verified and closed by either the submitter or the other reviewers. This is to ensure no bias and misjudgment by the author.

4.1.4 Define Specific Role Coverage for Reviewer

- Upstream reviewer
  - ❖ Roles: Software Manager, Project Lead, Quality Manager, etc.
  - ❖ Ensure the specification meets the customer's requirements.

- Downstream reviewer
  - ❖ Roles: Developer, Feature Architect, Peer, etc.
  - ❖ Ensure the software implementation meets the specification.

4.2 FTR Tool

To conduct an efficient FTR, besides the well-defined guidelines, an FTR tool is also important to assist and strengthen the FTR review efficiency which led the path to better quality release criteria. The organization should understand its needs and objectives to achieve in the project, then pick the right tool to accomplish the goals.

An effective FTR tool requires the following capabilities:

- Able to schedule the review meeting (Date, Time, Location, Path, etc.), assign the R&R for the participants accordingly, and send out the email notification.
- Able to capture all the defects/comments logged by the reviewers and arrange by review priority in one place.

- Able to capture all the defects/comments logged by the participants and review efforts in one place.
- Able to indicate which defects/comments need to be discussed or skip.
- Able to backtrack all the review records in one place.

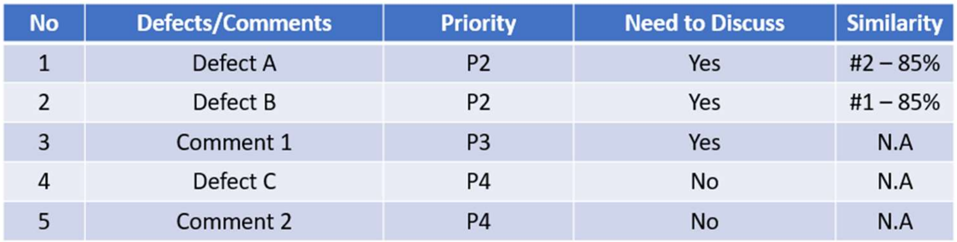
4.2.1 Types of FTR Tool

- Off the shelf
  - Commercial software tool which provides software solutions for the mass market.
  - Pros: Quick to implement, can try before buying, updates are included, support is included, etc.
  - Cons: Expensive and may be impossible to customize according to the needs, upgrade/support costs, etc.
- In-house development
  - Developing the software tool using your own company experts you have on hand.
  - Pros: Able to customize and suit the needs, less expensive, higher level of control, etc.
  - Cons: Slow to implement, lack of expertise, no upgrade or support if staff dismissal, etc.

A high-quality FTR tool should have the following features that will help make the work progress run more efficiently and help participants in the process quickly understand and align their thinking and the work to be done. FTR tool creates values by having a centralized repository for all the review comments, data consolidation, email notification, versioning & tagging, FTR dashboard provides users with the analysis capability by utilizing the consolidated review comments.

4.2.2 The Ability of Consolidation

The FTR tool should be able to group the defects or feedback based on severity (from highest to lowest), similarity (Help to identify duplication), and deprioritize the “don’t need to discuss” defects (Cosmetic issues like spelling mistakes or format errors). This will help to shorten the review meeting duration and only fully focus on those important items.

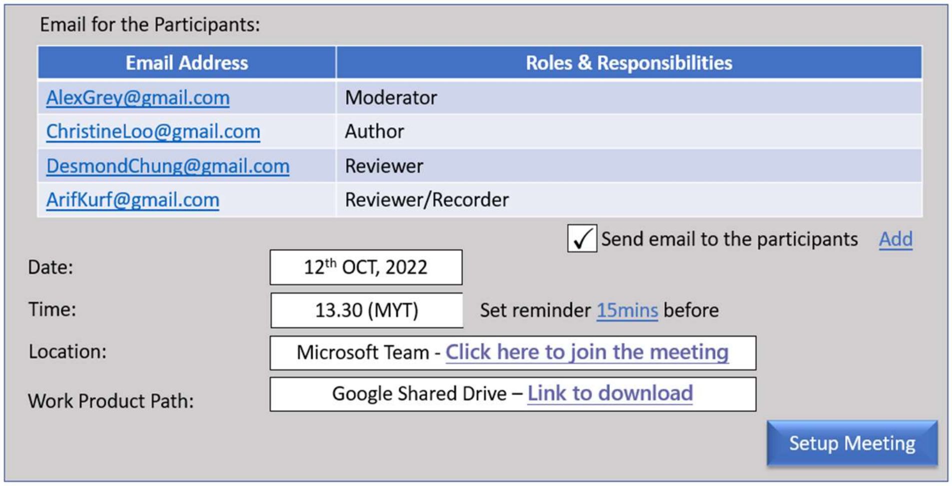


No	Defects/Comments	Priority	Need to Discuss	Similarity
1	Defect A	P2	Yes	#2 – 85%
2	Defect B	P2	Yes	#1 – 85%
3	Comment 1	P3	Yes	N.A
4	Defect C	P4	No	N.A
5	Comment 2	P4	No	N.A

Table-6 Defect Consolidation

4.2.3 The Ability of Notification

The FTR tool should be able to trigger an email notification when the review is scheduled. All new changes such as R&R, date, etc. should trigger another email notification as an update.



Email for the Participants:

Email Address	Roles & Responsibilities
AlexGrey@gmail.com	Moderator
ChristineLoo@gmail.com	Author
DesmondChung@gmail.com	Reviewer
ArifKurf@gmail.com	Reviewer/Recorder

☑ Send email to the participants Add

Date:	12th OCT, 2022	
Time:	13.30 (MYT)	Set reminder 15mins before
Location:	Microsoft Team - Click here to join the meeting	
Work Product Path:	Google Shared Drive – Link to download	

Figure-2 Email Notification

4.2.4 One-click Report Generation or One-glance Dashboard

The FTR tool should be able to present the work progress status (% open, % closed, % work in progress, % not an issue, etc.) via one-touch report generation or a one-glance dashboard. This will help to create quick insights or a summary of the overall status of project reporting.

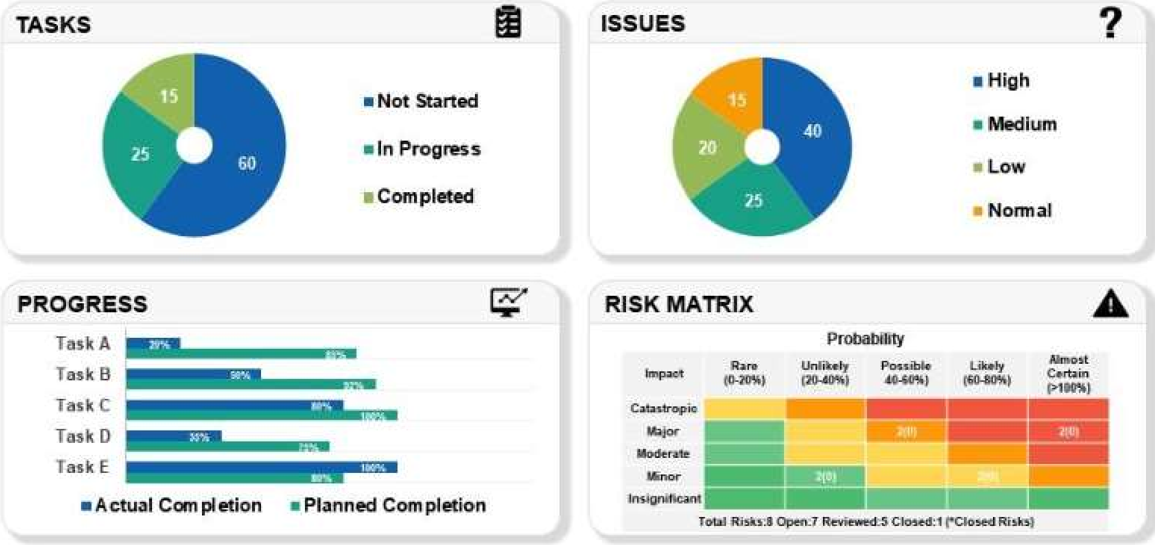
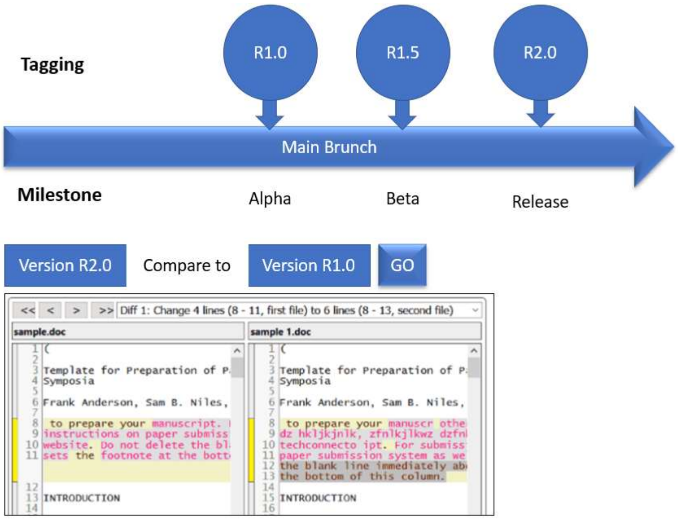


Figure-3 FTR Dashboard

4.2.5 Versioning and Tagging Capability

The FTR tool should be able to tag the versioning accordingly based on milestone release and be able to compare the delta between two different versions with just a few clicks. This will provide a quick summary of what has been changed between the current and previous releases.



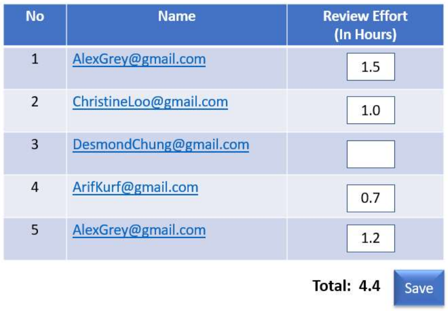
```
 1 (
 2
 3 Template for Preparation of P
 4 Symposia
 5
 6 Frank Anderson, Sam B. Niles,
 7
 8  to prepare your manuscript.
 9 instructions on paper submiss
10 website. Do not delete the bl
11 sets the footnote at the bott


12
13 INTRODUCTION
14
```

```
 1 (
 2
 3 Template for Preparation of P
 4 Symposia
 5
 6 Frank Anderson, Sam B. Niles,
 7
 8  to prepare your manuscr othe
 9 dz hkljkjnlk, zfnlkjlkwz dzfn
10 techconnecto ipt. For submiss
11 paper submission system as we
12 the blank line immediately ab
13 the bottom of this column.
14
15 INTRODUCTION
16
```

Figure-4 Versioning and Tagging

4.2.6 Log the Reviewing Efforts

The FTR tool should be able to capture all the reviewer's review duration. This will provide a reference point and help to estimate the total effort needed for a similar review in future.



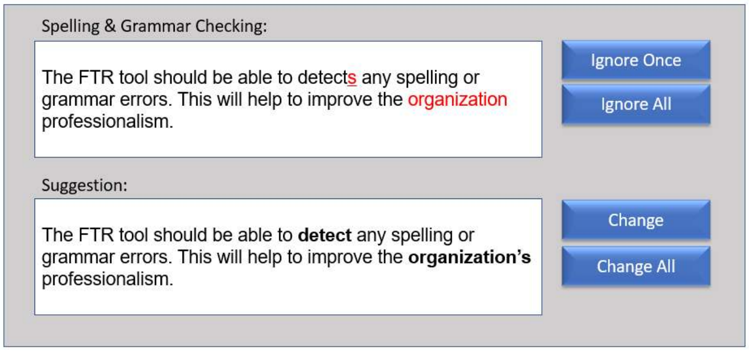
No	Name	Review Effort (In Hours)
1	AlexGrey@gmail.com	1.5
2	ChristineLoo@gmail.com	1.0
3	DesmondChung@gmail.com	
4	ArifKurf@gmail.com	0.7
5	AlexGrey@gmail.com	1.2

Total: 4.4

Figure-5 Time Logged for Review Efforts

4.2.7 Grammar & Spelling Checker

The FTR tool should be able to detect any spelling or grammar errors. This will help to improve the organization's professionalism.



Spelling & Grammar Checking:

The FTR tool should be able to detects any spelling or grammar errors. This will help to improve the organization professionalism.

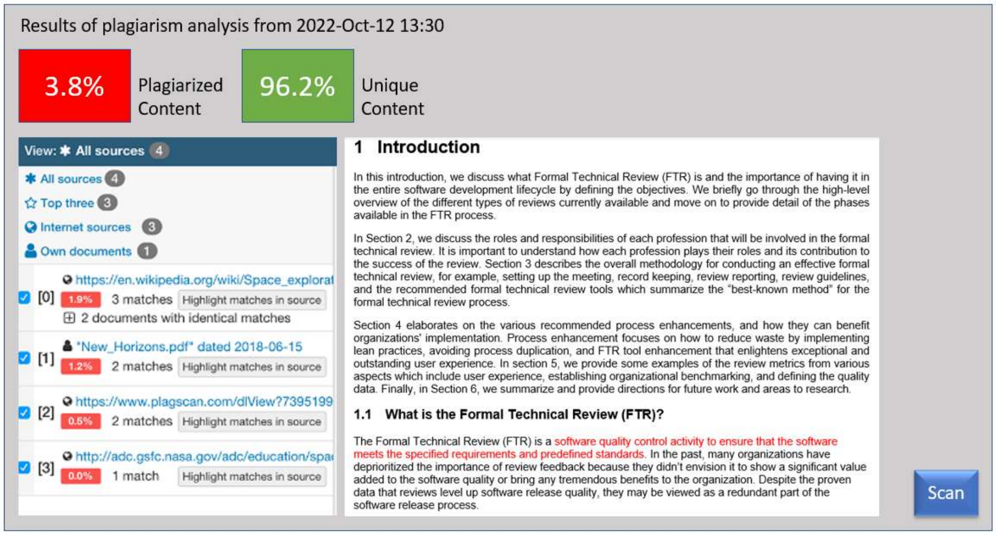
Suggestion:

The FTR tool should be able to **detect** any spelling or grammar errors. This will help to improve the **organization's** professionalism.

Figure-6 Grammar Check

4.2.8 Plagiarism Detection

The FTR tool should be able to detect plagiarism or copyright infringement within the work product. However, this is a ‘nice to have’ rather than an essential feature. This will help to avoid any legal repercussions or destroy the company's professional reputation.



Results of plagiarism analysis from 2022-Oct-12 13:30

3.8% Plagiarized Content

96.2% Unique Content

View: ✱ All sources 4

- ✱ All sources 4
- ☆ Top three 3
- Internet sources 3
- Own documents 1

☑ [0] https://en.wikipedia.org/wiki/Space_explorat
1.9% 3 matches
⊞ 2 documents with identical matches

☑ [1] "New_Horizons.pdf" dated 2018-06-15
1.2% 2 matches

☑ [2] https://www.plagscan.com/dlView?7395199
0.5% 2 matches

☑ [3] http://adc.gsfc.nasa.gov/adc/education/spa
0.0% 1 match

1 Introduction

In this introduction, we discuss what Formal Technical Review (FTR) is and the importance of having it in the entire software development lifecycle by defining the objectives. We briefly go through the high-level overview of the different types of reviews currently available and move on to provide detail of the phases available in the FTR process.

In Section 2, we discuss the roles and responsibilities of each profession that will be involved in the formal technical review. It is important to understand how each profession plays their roles and its contribution to the success of the review. Section 3 describes the overall methodology for conducting an effective formal technical review, for example, setting up the meeting, record keeping, review reporting, review guidelines, and the recommended formal technical review tools which summarize the “best-known method” for the formal technical review process.

Section 4 elaborates on the various recommended process enhancements, and how they can benefit organizations' implementation. Process enhancement focuses on how to reduce waste by implementing lean practices, avoiding process duplication, and FTR tool enhancement that enlightens exceptional and outstanding user experience. In section 5, we provide some examples of the review metrics from various aspects which include user experience, establishing organizational benchmarking, and defining the quality data. Finally, in Section 6, we summarize and provide directions for future work and areas to research.

1.1 What is the Formal Technical Review (FTR)?

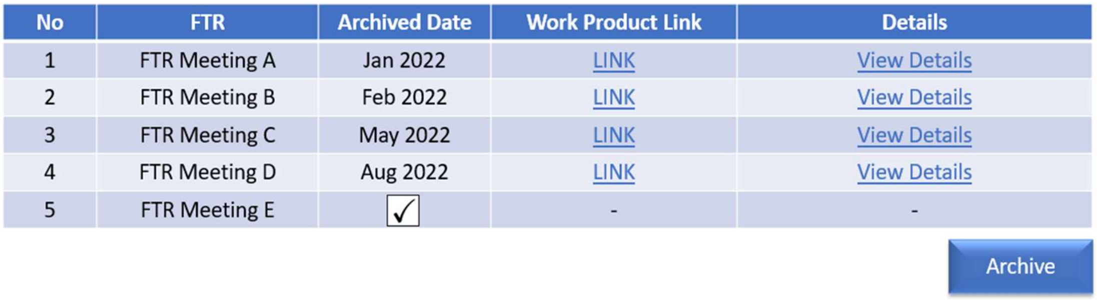
The Formal Technical Review (FTR) is a software quality control activity to ensure that the software meets the specified requirements and predefined standards. In the past, many organizations have deprioritized the importance of review feedback because they didn't envision it to show a significant value added to the software quality or bring any tremendous benefits to the organization. Despite the proven data that reviews level up software release quality, they may be viewed as a redundant part of the software release process.

Highlight matches in source
Highlight matches in source
Highlight matches in source
Highlight matches in source
Scan

Figure-7 Plagiarism Detection

4.2.9 Auto Archiving Capability

The FTR tool should be able to archive the work product and the review summary report automatically and have the capability to backtrack if needed. This will help to store as review evidence for quality auditing purposes.



No	FTR	Archived Date	Work Product Link	Details
1	FTR Meeting A	Jan 2022	LINK	View Details
2	FTR Meeting B	Feb 2022	LINK	View Details
3	FTR Meeting C	May 2022	LINK	View Details
4	FTR Meeting D	Aug 2022	LINK	View Details
5	FTR Meeting E	☑	-	-

Figure-8 Records Archiving

5 Review Metrics

A review can be considered effective only if the FTR identifies the gap at the right time. If it fails to achieve this will indicate that the review is not effective. Therefore, it is important to track and record gaps that do not get identified at the right time in a retrospective review.

Data collected for improvement purposes expresses progress over time and inspires the team to further improve, refining their products to achieve intended outcomes. Review data requires time to accumulate to define powerful metrics to strengthen the review process.

The recommended metrics are defined as follows:

- Mean time per issue logged.
- Total reviewers' efforts in a specific review.
- Total escaped issues by time.
- Total escaped issues by cost.
- Issue Priority.

The mean time per issue logged over review time metric measures the average time spent per issue logged per reviewer. Adding more reviewers to each review could expedite the review process but it depends on the reviewer's expertise and skillset.

The total reviewers' effort metric is to measure the effectiveness of the review process based on the total time spent and the total number of issues logged in the tool per reviewer. The greater number of reviewers being invited into the review may result in more review comments being logged.

The total escaped issues by time metric are to measure whether each assigned reviewer(s) has the expertise to perform an effective review. However, the work product with fewer defects does not mean the quality is excellent. It could be due to the reviewer who does not have sufficient knowledge or expertise in this review.

Issue priority metric, as the name implies, is to compare the issue arrival based on the priority. It is critical for the author to address and resolve the issues with the highest priority over the other issues.

As a result, having these metrics is to highlight how the review process might be adapted for future improvement based on the data collected and the trending from each of the metrics identified. The team should allocate sufficient time to analyze each metric, identify room for improvement in the FTR process and predict future trending.

Samples of the above-mentioned metrics and charts are shown below:

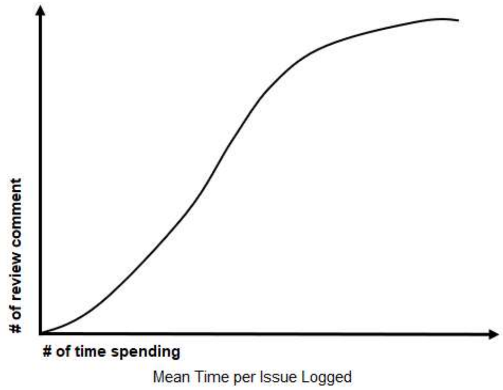


Figure-9 Mean Time per Issue Logged

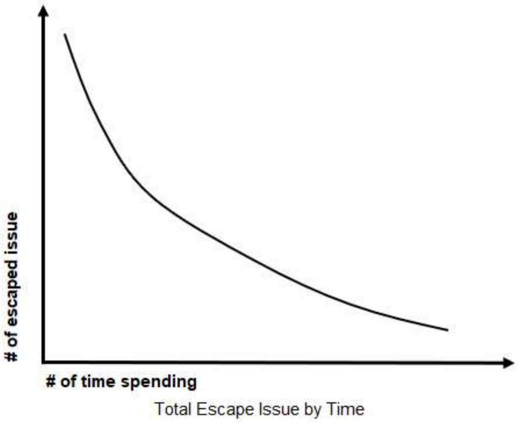


Figure-10 Total Escaped Issue by Time

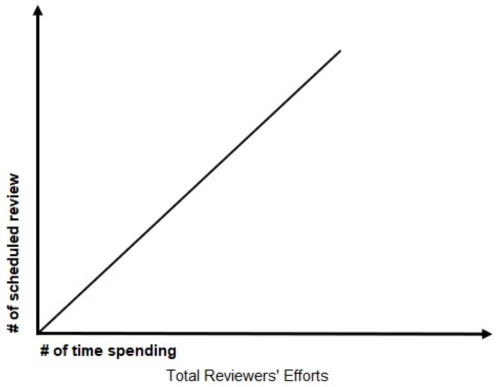


Figure-11 Total Reviewers' Efforts vs Scheduled Review

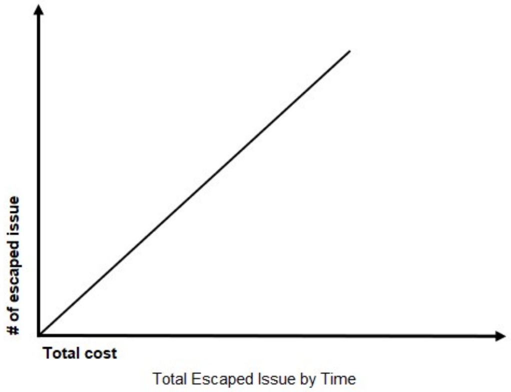


Figure-12 Total Escaped Issue by Cost

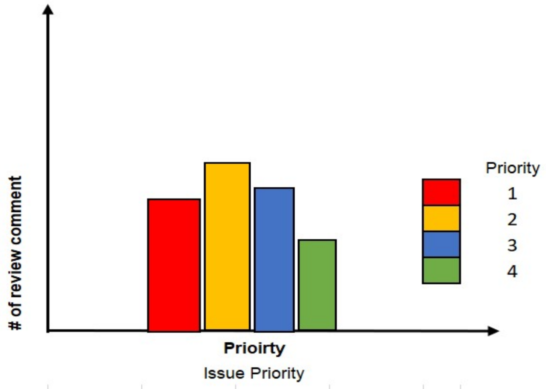


Figure-13 Issue Priority

Table-5 below shows the high-level summary of the FTR metrics based on the actual information captured on software architecture and design document review and improvement. In general, an effective FTR will adhere to the below guidance:

FTR Metrics	Improvement
Total defects captured =< 50	If the defects captured exceed the limit, it indicates that the work product is not ready. The FTR meeting should be called off, instead conducting one or more informal reviews as needed to ensure the total defects are reduced below the limit
Total reviewer time is =< 3 hours	Focus on small group of modules rather than the entire design
Escaped Issue =< 20%	Define clear roles and responsibilities for all participants and ensure the reviewers with correct expertise are invited into the review meeting to ensure the review coverage

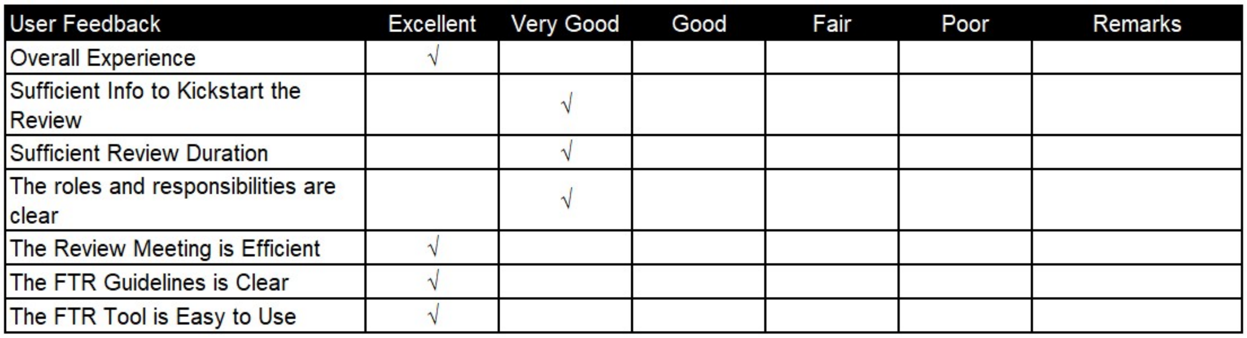
Table-7 Sample FTR Metrics

5.1 User Experience (UX)

Evaluating the user experience (UX) on the FTR tool and other FTR-related guidance at an organization is important and should not be missed. A strategy without metrics is like flying blind without indicators, hence clear metrics should be identified in our use case like review feedback, issue logged, and time spent in review. This is because these indicators represent the actual user experience that summarizes both good practices and the potential gaps.

The customer experience tracks the journey of touchpoints the customer may have with products and services. Similarly, the user experience includes all the touchpoints the employee has with the organization throughout their journey including onboarding, engagement, development, and growth. User experience metrics mainly focus on ease of use. Usability is familiar territory and something that user experience teams do well, so this makes sense as a starting point. The most used metrics are performance measures which we have described in section 5.0. These are objective measurements that record what people do.

The employee experience in FTR covered the review performance of the FTR guidelines and FTR tool in different areas of reviews throughout the Software Development Lifecycle (SDLC). The information gathered from the review indicators turns them into measurable metrics for further process improvement and enhancement. One of the methods to collect employee experience is to provide a short survey. For example:



User Feedback	Excellent	Very Good	Good	Fair	Poor	Remarks
Overall Experience	√					
Sufficient Info to Kickstart the Review		√				
Sufficient Review Duration		√				
The roles and responsibilities are clear		√				
The Review Meeting is Efficient	√					
The FTR Guidelines is Clear	√					
The FTR Tool is Easy to Use	√					

Table-8 User Feedback sample

Another source of data on the user experience might be creating a focus group of participants in a retrospective type of activity to reflect on the FTR process so that feedback trends or abnormalities can be further analyzed.

5.2 Benchmarking

Benchmarking is an organizational tool to drive continuous improvements using best practices. This can translate into increased efficiency and create competitive advantages. Data collection from the FTR process may help organizations to develop a process for benchmarking their FTR work by accumulating and comparing the data and turning it into an acceptable value. Establishing the benchmark values for each metric of the review process is critical to ensuring the effectiveness of the review performance for each review meeting. However, the benchmark values vary from one review to another review and are highly dependent on the complexity of the software deliverables. For most cases, an effective review meeting should have a minimum number of review comments logged as part of the process despite where the issues coming from.

For instance, the review metrics are built on top of the benchmark values and provide an initial measurement of the review process. Gap analysis is performed when the actual review output is mismatched to the benchmark values and could be far from the acceptable range. A deep dive into root cause analysis (RCA) should be carried out when the review's metrics diverge from the benchmark and continuously drives further process improvement. It is recommended to continue to fine-tune the benchmark values so more accurate benchmark values can be archived.

5.3 Quality Data

The below information is captured based on the actual FTR meeting from the Feature requirement. The chart shows a total of 3 FTR meetings conducted for 3 new features named FTR1, FTR2, and FTR3.

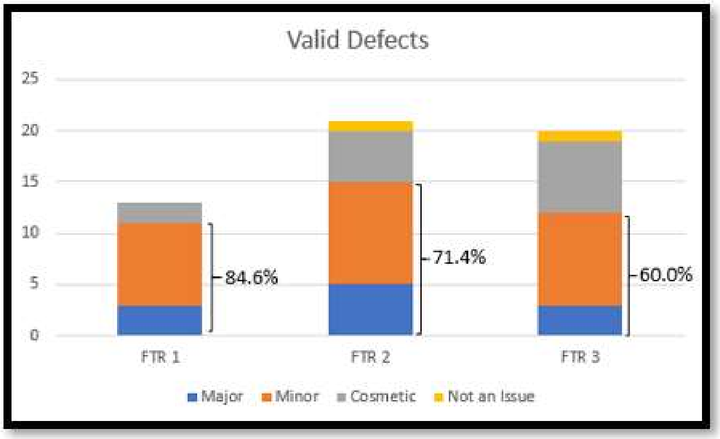


Figure-14 Valid Defects Summary on FTRs

Formal Technical Review	Major	Minor	Cosmetic	Not an Issue	Valid Defects (%)
FTR 1	3	8	2	0	84.60%
FTR 2	5	10	5	1	71.40%
FTR 3	3	9	7	1	60.00%

Table-9 Valid defects are based on the sum of Major and Minor

Based on the actual data captured from the FTR review from different projects, we can confidently set a disclaimer that without an FTR review being conducted, at least 60% of the valid defects might escape without awareness and release to market. Consequently, an additional cost will be imposed to address the escaped defects which will increase the overall cost of the project for issue fixing and introduce a poor-quality impression to the customer. In the nutshell, prevention is always better than cure, it's easier to stop the defects happening in the earlier stage than to repair the damage after it has happened.

6 Conclusion

In this paper, we provide a detailed explanation of the FTR which include the overview, objectives, types of reviews, roles & responsibilities, etc. Meanwhile, we also listed the recommendations for the FTR enhancements which we believe can elevate the FTR capability to a new level. The key benefits of FTR to the organization are clear: to improve the work product quality and identify the potential defects upfront to avoid any unnecessary costs being imposed in post-release defect fixing. In short, to develop an organization with strong quality culture, we must adopt new changes and view them as an improvement opportunity. It is more about managing costs more intelligently. We need to invest the time upfront to do these analyses up front, rather than spend a lot more time doing rework when defects are identified late in the validation process. FTR has proven its ability, and now it is up to you to still view it as a redundant process or a quality improvement opportunity.

References

1. Design for instrumentation: high quality measurement of formal technical review, https://link.springer.com/article/10.1007/BF02420943
2. Measuring Effectiveness of a Review, https://www.ijsrp.org/research-paper-0315/ijsrp-p3969.pdf
3. Three Employee Experience Must-Have Metrics, https://elearningindustry.com/three-employee-experience-must-have-metrics
4. Formal Technical Review in Software Engineering, https://www.geeksforgeeks.org/formal-technical-review-ftr-in-software-engineering/?ref=gcse
5. Traceability of Implementation to Design and Requirements Specs: A FTR Method (Reverse Engineering Tool), https://www.computerscijournal.org/vol8no2/traceability-of-implementation-to-design-and-requirements-specifications-a-formal-technical-review-method-reverse-engineering-tool/
6. Different Phases of Formal Review, https://www.geeksforgeeks.org/different-phases-of-formal-review/
7. Roles and Responsibilities in Review, https://www.geeksforgeeks.org/roles-and-responsibilities-in-review/#:~:text=Formal%20review%20generally%20provides%20various,early%20in%20software%20development%20process
8. Difference between Software Inspection and Technical Review, https://www.geeksforgeeks.org/difference-between-software-inspection-and-technical-review/?ref=gcse
9. Software Review and Formal Technical Review, https://mechomotive.com/software-review-and-formal-technical-review/