

2. Software Verification

Page No.	
Date	

* Verification Methods

1) peer Reviews

- This is the simplest way of reviewing the documents/ programs to find out faults during Verification
- we give the document(s) / program(s) to someone else & ask to review the documents / pgm's .
- we expect views about the quality of the document & also expect to find faults.
This type of informal activity may give very good results without spending any significant resources.
- The reviewer may prepare a report of observation and findings or may inform verbally during discussion.
- This is an informal activity to be carried out by peers & may be very effective if reviewers have domain knowledge, good programming skills & proper involvement.

2) Walkthroughs:

- walkthroughs are more formal & systematic than peer reviews.
- In a walkthrough, the author of the document presents the document to a small group of two to seven persons.
participants are not expected to prepare anything

only the presenter, who is the author, prepares for the meeting.

- After the review, the author writes a report about findings & any faults pointed out in the meeting.
- The disadvantages of this system are the non-preparation of participants & incompleteness of the document presented by author.
- The participants may not be able to ask many penetrating questions.
- It may help us to find potential faults & may also be used for sharing the documents with others.

3) Inspections.

- This is the most formal type of verification method and is commonly known as inspections.
- These are different from peer reviews & walkthroughs.
- The presenter is not the author but some other person who prepares & understands the document being presented.
- This forces that person to learn & review that document prior to the meeting.
- The document(s) is / are distributed to all participants in advance in order to give them sufficient time for preparation.
- Rules for such meetings are fixed and communicated to all participants.

4) Applications

All 3 verification methods are popular and have their own strengths & weaknesses. These methods are compared on specific issues & this comparison is given in table.

SR. No	Method	Presenter	No. of participants. preparation	Major	Report	Strengths	Weakness
1. Peer Reviewers	No one	1 or 2	Not Required	optional	Inexpensive but find faults	Independent ability of team	Opisde-
2. walkthrough Author	Author	2 - 7	Only presenter is required to be prepared	prepared by presenter	knowledge sharing	Find faults	Faults noted EXPENSIVE
3. Inspection.	Someone other than author	3-6	All participants are required to be prepared	prepared by moderator	effective & find many faults.	Expensive & requires participation.	

* Software Requirements Specification (SRS) document verification.

- The outcome of the first phase of the s/w development life cycle is the SRS document.
- The SRS document should cover both functional requirements & non-functional requirements.
Functional requirements are the expectations from the proposed software. They are also known as product features.
Nonfunctional requirements are quality requirements that stipulate how well the s/w does what it has to do.

1) Nature of the SRS document.

The SRS should include the following:

- (i) Expectations from the software
- (ii) Interfaces of the software.
- (iii) Non-functional Requirements.
- (iv) Implementation difficulties & limitations.

The SRS writer(s) should not include design & implementation details. It should be written in simple, clear & unambiguous language which may be understandable to all developers & customers.

2) characteristics and organization of the SRS Document.

The SRS document acts as a contract between the developer and customer. This document should have the following characteristics as given in IEEE recommended practice for software requirements specifications (IEEE std 830_1998).

- The SRS document is reviewed by the testing person(s) by using any verification method.
- A checklist is a popular verification tool which consists of a list of critical information content that a deliverable should contain.
- An SRS document checklist should address the following issues :

(i) correctness

Every requirement stated in the SRS should correctly represent an expectation from the proposed system.

(ii) Ambiguity

There may be an ambiguity in a stated requirement. If a requirement conveys more than one meaning, it is a serious problem. Every requirement must have a single interpretation only.

(iii) completeness

The SRS document should contain all significant functional requirements and

non-functional requirements.

The completeness of the SRS document must be checked thoroughly by a checklist.

(iv) consistency

Consistency of the document may be maintained if the stated requirements do not differ with other stated requirements within the SRS documents.

(v) Verifiability

The SRS document is said to be verifiable, if and only if, every requirement stated there in is verifiable.

Non-verifiable requirements include statements like 'good interfaces', 'usually', 'well', etc. These statements should not be used.

VI) Modifiability

The SRS document should incorporate modifications without disturbing its structure & style. Thus, changes may be made easily, completely and consistently while retaining the framework. modifiability is a very important characteristic due to frequent changes in the requirements.

vii) Traceability

The SRS document is traceable if the origin of each requirement is clear & may also help for future development. It may help to structure the document & should find place in the design of the checklist.

viii) Feasibility

Some of the requirements may not be feasible to implement due to technical reasons / lack of resources. Such requirements should be identified & accordingly removed from the SRS document.

* Software Design Description (SDD) Document Verification.

- We prepare the SDD document from the SRS document.
- The SDD should address all design entities along with their attributes.

1. Organisation of the SDD Document.

- We have the IEEE recommended practice for software design description, which is a popular way to organise an SDD document.
- The organisation of SDD is given as per IEEE STD 1016 - 1998.

- two popular design techniques are function oriented design and object oriented design
- we may use any approach depending on the nature and complexity of the project
- The verification process may be carried out many times in order to improve the quality of the SDD.
- The SDD provides a bridge bet'w requirements and implementation.

2. The SDD Document checklist

- SDD document verification checklist may provide opportunities to reviewers for focusing on important areas of the design.
- The SDD design starts as a process for translating requirements stated in the SRS documenting a user-oriented functional design.
- A checklist may help to structure the design review process. There are many ways to design a checklist which may vary with the nature, scope, size & complexity of the project.
- However, organisations may modify this checklist depending on SDEVNQG. practice and type of the project.

* Source code Reviews.

A source code review involves one / more reviewers examining the source code & providing feedback to the developers, both positive & negative.

I. ISSUES Related To Source code Review.

- Always use meaningful variables
- Avoid confusing words in names, ~~not~~
- declare local variables & avoid global variables to the extent possible.
- minimize the visibility of variables
- Do not overload variables with multiple meanings
- Define all variables with meaningful, consistent & clear names.
- Do not unnecessarily declare variables
- use comments to increase the readability of the source code.
- Generally, comments should describe what the source code does & not how the source code works.
- Always update comments while changing the source code
- use spaces & not tabs
- All drivers should be tested for a/garbage val.
- Always remove unused lines of the source code.
- minimize the module coupling & maximize the module strength.

2. checklist of source code reviews

- A checklist should at least address the above issues.
- A generic checklist is given.
- we may also prepare a programming language specific checklist which may also consider the specific language issues.

* User documentation verification.

- = User documentation may be provided as a user manual in electronic form, as a printed booklet / in the form of online help.

1. Review process issues.

These documents should be reviewed thoroughly & proper consistency should be maintained in all documents.

- The documents should be written in simple clear & short sentences.
- Installation procedure of the software must be explained step by step with proper justifications.
- All tables, figures & graphs should be numbered properly.

2. USE documentation checklist

A checklist always helps the review process.

General Issues

- Is the document easy to read?
- Is the document easy to understand?
- _____ II _____ well organized?
- Does the document look professional?
- Are spellings & grammar correct?
- Are all references properly placed in text?

* Software project Audit.

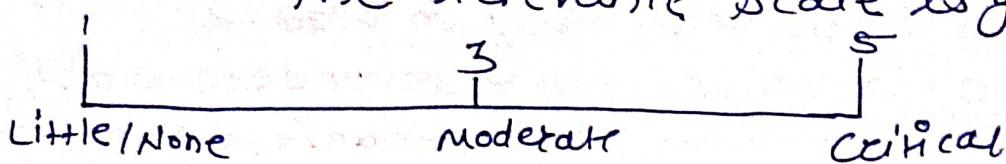
Audit of a software project is a very important activity & may be carried out at any time during the software development life cycle.

Generally, auditors are appointed by the top management to review the progress of the project. The auditors are different from the developers and testers and may not have any involvement in the project.

1. Relevance scale

A relevance scale has been given in project audit & review checklist to measure the relevance of any attribute at the time of auditing the project.

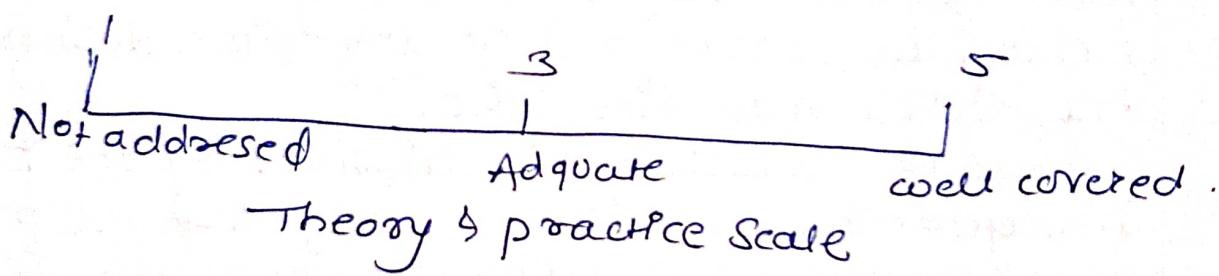
many attributes have been identified in the checklist. We have to find their relevance to the project at the stage when the audit is being conducted. The relevance scale is given as:



Relevance scale when the audit is conducted

2. Theory and practice Scale.

We may have to further indicate the strengths & weaknesses of the attributes given in project audit & review checklist, in theory & practice on the scale as given below:



Theory & practice scale

3. Project Audit & Review checklist.

- This checklist has been designed by ICAI bank which has been used by many organizations.
- All activities are reviewed on the basis of its relevance & strength/weakness at any point of time.
- Relevance scale & theory-practice scale may help us to understand the status of various attributes.

U
28/12/2017