SE ZG501
Software Quality Assurance and Testing
Session 7

Practices - to develop an effective and efficient personal review

- Pause between the development of a software product and its review.

- Examine products in hard copy rather than electronically.

- Check each item on the checklist once completed.

- Update the checklists periodically to adjust to your personal data.

- Build and use a different checklist for each software product;

- Verify complex or critical elements with an in depth analysis.

ENTRY CRITERIA
- None

INPUT
- Software product to review

ACTIVITIES
1. Print:
- Checklist for the software product to be reviewed
- Standard (if applicable)
- Software product to review
2. Review the software product, using the first item on the checklist and cross this item off when the review of the software product is completed
3. Continue review of the software product using the next item on the checklist and repeat until all the items in the list have been checked
4. Correct any defects identified
5. Check that each correction did not create other defects.

EXIT CRITERIA
- Corrected software product

OUTPUT
- Corrected software product

MEASURE
- Effort used to review and correct the software product measured in person-hours with an accuracy of +/−15 minutes.
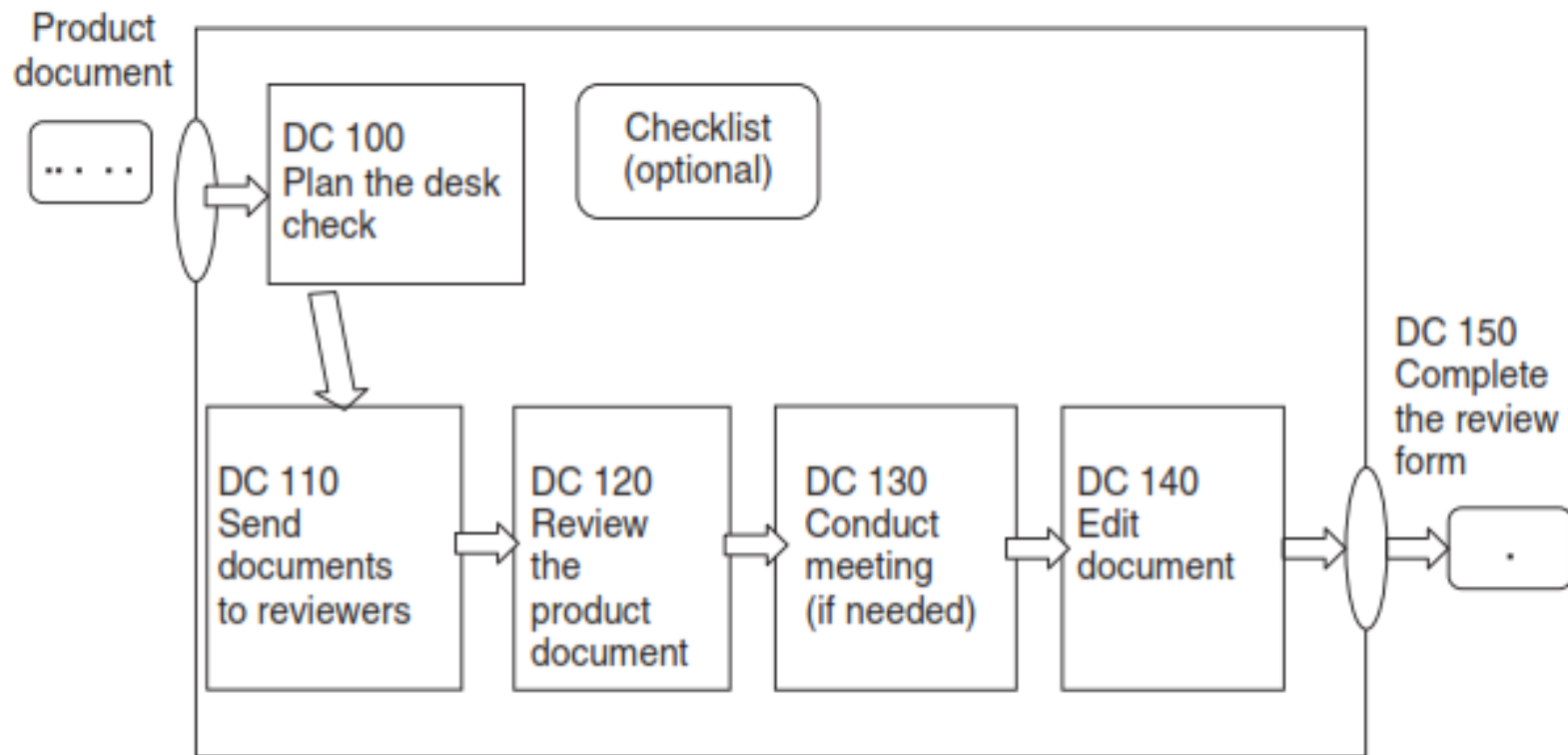
Figure 5.6 Personal review process.

Source: Adapted from Pomeroy-Huff et al. (2009) [POM 09].

Desk-Check Reviews

- A type of peer review that is not described in standards is the desk-check review (Pass around) .

Figure 5.7 Desk-check review.

There are six steps.

- Initially, the author plans the review by identifying the reviewer(s) and a checklist.

- A checklist is an important element of a review as it enables the reviewer to focus on only one criterion at a time.

- A checklist is a reflection of the experience of the organization.

- Then, individuals review the software product document and note comments on the review form provided by the

- author.

- When completed, the review form can be used as "evidence" during an audit.

Important features of checklists:

– Each checklist is designed for a specific type of document (e.g., project plan, specification document).

– Each item of a checklist targets a single verification criteria.

– Each item of a checklist is designed to detect major errors. Minor errors, such as misspellings, should not be part of a checklist.

– Each checklist should not exceed one page, otherwise it will be more difficult to use by the reviewers.

– **E**ach checklist should be updated to increase efficiency.

– Each checklist includes a version number and a revision date.

- The following text box presents a generic checklist, that is, a checklist that can be used for almost any type of document to be reviewed (e.g., project plan, architecture).

- For each type of software product (e.g., requirements or design), a specific checklist will be used.

Generic Checklist

LG 1 (COMPLETE). All pertinent information should be included or referenced.

LG 2 (RELEVANT). All information must be relevant to the software product.

LG 3 (BRIEF). Information must be stated succinctly.

LG 4 (CLEAR). Information must be clear to all reviewers and users of the document.

LG 5 (CORRECT). Information does not contain errors.

LG 6 (COHERENT). Information must be consistent with all other information in the document and its source document(s).

LG 7 (UNIQUE). Ideas must be described once and referenced afterward.

Adapted from Gilb and Graham (1993) [GIL 93]

- In the **third step of the desk-check process**, the reviewers verify the document and record their comments on the review form.

- The **author reviews the comments as part of step 4**.

- If the author agrees with all the comments, he incorporates them into his document. **After this meeting, one of three options should be considered:** the comment is incorporated as is, the comment is ignored, or it is incorporated with modifications.

Next step, the author can make the corrections and note the effort spent reviewing and correcting the document, that is, the time spent by the reviewers as well as the time spent by the author to correct the document and conduct the meeting if this is the case.

In the final step, the author completes the review form illustrated in Figure 5.9.

ENTRY CRITERIA
- The document is ready for a review

INPUT
- Software product to review

DC 100. Plan the Desk-Check

Author:
- Identifies reviewers
- Chooses the checklist(s) to use
- Completes the first part of the review form

DC 110. Send documents to reviewers

Author:
- Provides the following documents to the reviewers:
    - Software product to review
    - Review form
    - Checklist(s)

DC 120. Review the software product

Reviewers:
- Check the software product against the checklist
- Complete the review form with
    - comments
    - effort to conduct the review
- Sign and return the form to the author

DC 130. Call a meeting (if needed)

Author:
- Reviews the comments
    - If the author agrees with all the comments, they are incorporated in the software product
    - If the author does not agree with all the comments, or believes some comments have a significant impact, then the author:
        - Convenes a meeting with the reviewers
        - Leads the meeting to discuss the comments and determine course of action:
            - Incorporate the comment as is
            - Ignore the comment
            - Incorporate the comment with modifications

DC 140. Correct the software product
The author incorporates the comments received.

DC 150. Complete the review form

Author:
- Completes the review form with:
    - Total effort (i.e., by all the reviewers) required to review the software product
    - Total effort required to correct the software product
- Signs the review form

EXIT CRITERIA
- Corrected software product

OUTPUT
- Corrected software product
- Completed and signed review form

MEASURE
- Effort required to review and correct the software product (person hours).

Figure 5.8   Desk-check review activities.

Desk check review form

Name of author: _____				Review date (yyyy-mm-dd): _____	
Document title: _____				Reviewer name: _____	
Document version: _____					
Comment No.	Document page	Line # / location	Comments	Disposition of comments*	Remarks
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					

Disposition of comment: Inc: Incorporate as is;   NOT: Not incorporate,   MOD: Incorporate with modification

Effort to review document (hour): _____
Effort to correct document (hour): _____

Signature of reviewer: _____

Signature of author: _____

Figure 5.9   Desk-Check review form.

The IEEE 1028 Standard

The **IEEE 1028-2008 Standard** defines **five types of software reviews and audits**, detailing their purpose and process.

It provides guidelines on **how to conduct systematic reviews** during **software acquisition, development, operation, and maintenance** to ensure quality and compliance.

✅ Defines **what to review** and **how to review it**
✅ Ensures **software quality, correctness, and reliability**
✅ Covers **processes for structured reviews and audits**

This standard provides descriptions of the particular types of reviews and audits included in the standard as well as tips.

a) Introduction to review: describes the objectives of the systematic review and provides an overview of the systematic review procedures;

b) Responsibilities: defines the roles and responsibilities needed for the systematic review.

c) Input: describes the requirements for input needed by the systematic review;

d) Entry criteria: describes the criteria to be met before the systematic review can begin, including the following:
   1) Authorization,
   2) Initiating event;

e) Procedures: details the procedures for the systematic review, including the following:
   1) Planning the review;
   2) Overview of procedures;
   3) Preparation;
   4) Examination/evaluation/recording of results;
   5) Rework/follow-up;

f) Exit criteria: describe the criteria to be met before the systematic review can be considered complete;

g) Output: describes the minimum set of deliverables to be produced by the systematic review.

IEEE 1028 -The types of reviews and audits

1. **Management Review**
   - Evaluates project **progress, schedules, and requirements.**
   - Conducted by management to ensure alignment with goals.

2. **Technical Review**
   - Assesses software for **functional suitability and compliance.**
   - Performed by technical experts to detect specification gaps.

3. **Inspection**
   - A **formal, structured** review to **identify errors and deviations.**
   - Involves step-by-step checks against standards.

4. **Walkthrough**
   - The author **presents the software** to the team.
   - Participants **ask questions, give feedback, and suggest improvements.**

5. **Audit**
   - An **independent assessment** for **standards and contract compliance.**
   - Ensures the software meets regulatory and quality requirements.

WALK-THROUGH

- The purpose of a walk-through is to evaluate a software product.

- A walk-through can also be performed to create discussion for a software product"

- Objectives of the walk-through:
  - Find anomalies.
  - Improve the software product.
  - Consider alternative implementations.
  - Evaluate conformance to standards and specifications.
  - Evaluate the usability and accessibility of the software product.

Usefulness of a Walk-Through

- Identify errors to reduce their impact and the cost of correction.
- Improve the development process.
- Improve the quality of the software product.
- Reduce development costs.
- Reduce maintenance costs.

Table: Characteristics of Reviews and Audits (IEEE 1028 Standard)

Characteristic	Management Review	Technical Review	Inspection	Walk-through	Audit
Objective	Monitor progress	Evaluate conformance to specifications and plans	Find anomalies, verify resolution, ensure product quality	Identify anomalies, examine alternatives, improve product	Independently verify compliance with standards and regulations
Recommended Group Size	2 or more people	2 or more people	3-6	2-7	1-5
Volume of Material	Moderate to High	Moderate to High	Relatively low	Relatively low	Moderate to High
Leadership	Responsible manager	Lead engineer	Trained facilitator	Facilitator or author	Lead auditor
Management Participation	Yes	Sometimes, if evidence is required	No	No	No, but may be called for evidence
Output	Management review documentation	Technical review documentation	Anomaly list, summary, inspection report	Anomaly list, action items, follow-up proposals	Formal audit report, findings, deficiencies

INSPECTION REVIEW

This section briefly describes the inspection process that Michael Fagan developed at IBM in the 1970s to increase the quality and productivity of software development.

The purpose of the inspection, according to the IEEE 1028 standard, is to detect and identify anomalies of a software product including errors and deviations from standards and specifications.

Throughout the development or maintenance process, developers prepare written materials that unfortunately have errors. It is more economical and efficient to detect and correct errors as soon as possible. Inspection is a very effective method to detect these errors or anomalies.

According to the IEEE 1028 standard, inspection allows us to

a) verify that the software product satisfies its specifications;

b) check that the software product exhibits the specified quality attributes;

c) verify that the software product conforms to applicable regulations, standards, guidelines, plans, specifications, and procedures;

d) identify deviations from provisions of items (a), (b), and (c);

e) collect data, for example, the details of each anomaly and effort associated with their identification and correction;

f) request or grant waivers for violation of standards where the adjudication of the type and extent of violations are assigned to the inspection jurisdiction;

g) use the data as input to project management decisions as appropriate (e.g., to make trade-offs between additional inspections versus additional testing).

Major steps of the inspection process, Each step is composed of a series of inputs, tasks and outputs.
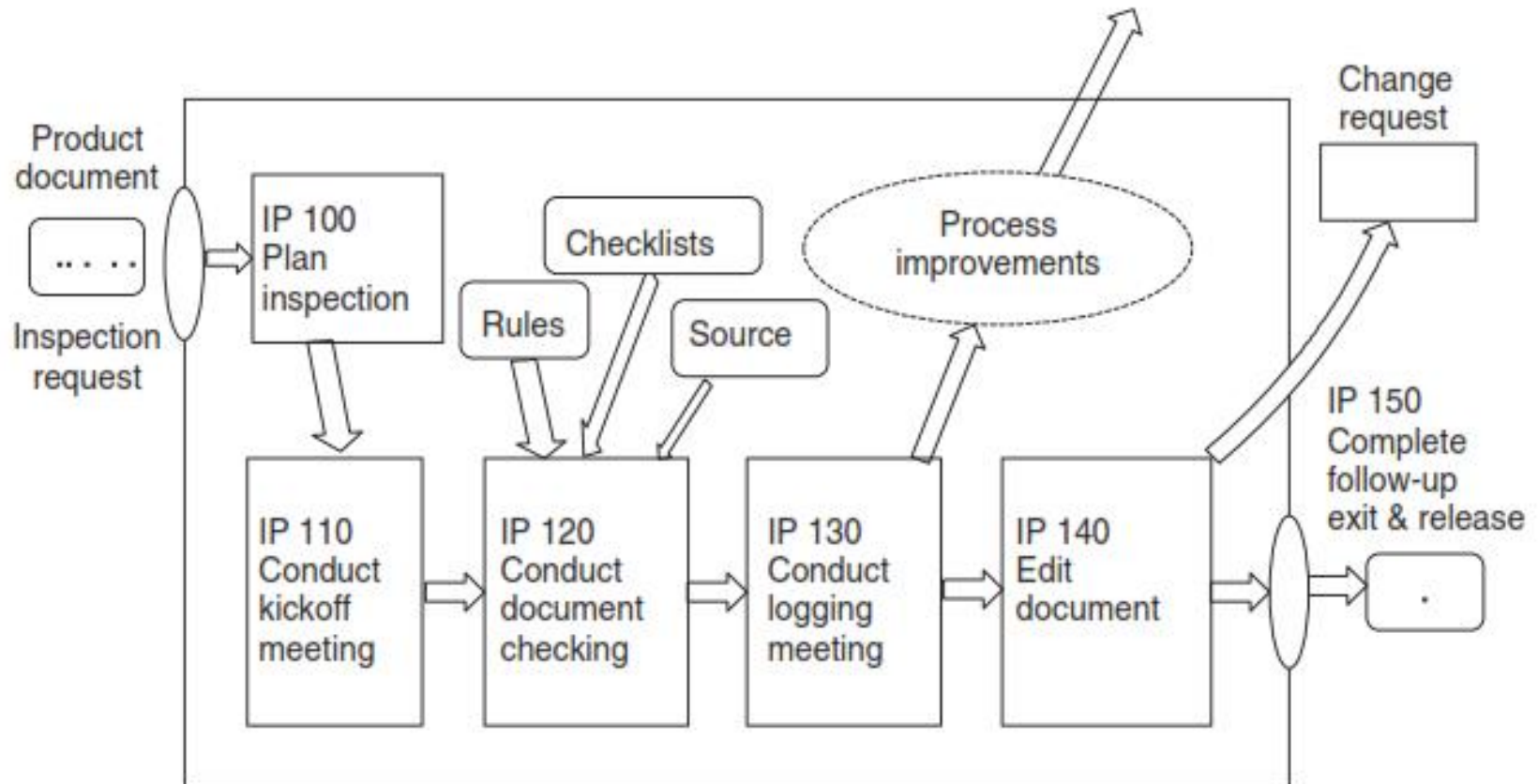
Figure 5.12   The inspection process.

PROJECT LAUNCH REVIEWS AND PROJECT ASSESSMENTS

In the SQAP of their projects, many organizations plan a project launch or kick-off meeting as well as a project assessment review, also called a lessons learned review.

QUIZ

1. What is the main purpose of a **desk-check review**?

   A) To test software functionality.

   B) To pass around a document for individual review and

   collect feedback.

   C) To perform automated code testing.

   D) To execute the software and detect runtime errors.

What is the main purpose of a **desk-check review**?

A) To test software functionality.

B) To pass around a document for individual review and

collect feedback.

C) To perform automated code testing.

D) To execute the software and detect runtime errors.

Answer: B) To pass around a document for individual review and collect feedback

QUIZ

2. What are the **major benefits** of conducting an **inspection review**? *(Select all that apply.)*

A) Detect and identify anomalies early in the process
B) Improve software quality
C) Reduce development and maintenance costs
D) Automate software debugging

What are the **major benefits** of conducting an **inspection review**? *(Select all that apply.)*

A) Detect and identify anomalies early in the process
B) Improve software quality.
C) Reduce development and maintenance costs.
D) Automate software debugging.

Correct Answers: A, B, C

Project Launch Review

The project launch review is a management review of: the milestone dates, requirements, schedule, budget constraints, deliverables, members of the development team, suppliers, etc.

Some organizations also conduct kick-off reviews at the beginning of each of the major phases of the project when projects are spread over a long period of time

Before the start of a project, team members ask themselves the following questions:

- Who will the members of my team ?

- Who will be the team leader?

- What will my role and responsibilities ?

- What are the roles of the other team members and their responsibilities?

- Do the members of my team have all the skills and knowledge to work on this project?

MEASURES

Measures are mainly used to answer the following questions:

- How many reviews were conducted?
- What software products have been reviewed?
- How effective were the reviews (e.g., number of errors detected by number of hours for the review)?
- How efficient were the reviews (e.g., number of hours per review)?
- What is the density of errors in software products?
- How much effort is devoted to reviews?
- What are the benefits of reviews?

The measures that allow us to answer these questions are:

- Number of reviews held.

- Identification of the revised software product.

- Size of the software product (e.g., number of lines of code, number of pages);

- Number of errors recorded at each stage of the development process;

- Effort assigned to review and correct the defects detected.

SELECTING THE TYPE OF REVIEW

To determine the type of review and its frequency, the criteria to be considered are:

- The risk associated with the software to be developed.

- The criticality of the software.

- Software complexity.

- The size and experience of the team,

- The deadline for completion, and software size.

Table 5.5   Example of a Matrix for the Selection of a Type of Review

Product	Technical drivers—complexity		
	Low	Medium	High
Software requirements	Walk-through	Inspection	Inspection
Design	Desk-check	Walk-through	Inspection
Software code and unit test	Desk-check	Walk-through	Inspection
Qualification test	Desk-check	Walk-through	Inspection
User/operator manuals	Desk-check	Desk-check	Walk-through
Support manuals	Desk-check	Desk-check	Walk-through
Software documents, for example, Version Description Document (VDD), Software Product Specification (SPS), Software Version Description (SVD)	Desk-check	Desk-check	Desk-check
Planning documents	Walk-through	Walk-through	Inspection
Process documents	Desk-check	Walk-through	Inspection

Tools for conducting Reviews

- **dutils** – Finds and organizes program identifiers.
- **Egrep** – Searches text using patterns.
- **Find** – Locates files on a system.
- **Diff** – Compares files to find differences.
- **Cscope** – Helps browse C code.
- **LXR** – Shows source code online with cross-referencing.

AUDITS

Audits are one of the most formal types of reviews in software quality assurance.

Different types of audits serve **different purposes**, such as:

- Ensuring a software company follows industry standards.

- Verifying that a supplier meets a client's requirements.

The **cost and independence of the auditor** depend on the type of audit being conducted.

This helps organizations **maintain compliance, improve quality, and build trust** with clients.

Key Components of a Software Audit

Management System – Defines policies, objectives, and processes for quality compliance.

Audit Criteria – Standards or policies used to evaluate compliance (e.g., ISO 9001, CMMI).

Audit Evidence – Documents, reports, and test results proving adherence to audit criteria.

Audit Process – Steps include planning, evidence collection, evaluation, reporting, and corrective actions.

Internal audits

- Called first party audits

- Conducted by the organization itself, or on its behalf, for management review and other internal purposes (e.g. to confirm the effectiveness of the management system or to obtain information for the improvement of the management system).

- Internal audits can form the basis for an organization's self-declaration of conformity.

External audits

- Includes second and third party audits.

- Second party audits are conducted by parties having an interest in the organization, such as customers, or by other persons on their behalf.

- Third party audits are conducted by independent auditing organizations, such as regulators or those providing certification.

There are different types of audit:

– audits to verify the compliance to a standard, such as the audits described in International Organization for Standardization (ISO) standards such as ISO

9001 and IEEE 1028;

– Compliance audits for a model such as the Capability Maturity Model Integration (CMMI) that are used to select a supplier before awarding a contract or assess a supplier during a contract;

– Audits ordered by the management team of the organization to verify the progress of a project against its approved plan.

Why audit?

Software project audits are usually requested by management to ensure that the software team and contracted suppliers:

- know their duties and obligations toward the public, their employers, their customers, and their colleagues;

- use the processes, practices, techniques, and methods suggested by the company;

- reveal any deficiencies and shortcomings in daily operations and try to identify required corrective actions (CAs);

- are encouraged to develop a personal training plan for their professional skills;

- are monitored in the course of their work on high profile projects of the company.

TYPES OF AUDITS

Internal Audit

- A **first-party audit** is an internal audit conducted by an organization to assess its own processes,
- First-party audit, can be useful for a software supplier wanting to obtain an ISO 9001 certification.
- It is the least expensive approach to prepare for conformity to an international standard.

Second-Party Audit

- conducted by parties having an interest in the organization, such as customers, or by other persons on their behalf.

Third party audits are conducted by independent auditing organizations, such as regulators or those providing certification.

- ISO does not provide certification services or issue certificates.

- Instead, certification bodies use standards like ISO 19011 for auditing guidelines and ISO/IEC 17021-1 for certification requirements.

- These standards ensure compliance with certifications like ISO 9001.

Project Assessment and Control Process

- **Align and Validate Plans:** Ensure project **plans are feasible and aligned with business objectives**.

- **Monitor Progress:** Assess the **current status of the project, technical performance, and process efficiency**.

- **Guide Execution:** Ensure the **project stays on schedule, within budget, and meets technical goals**.

- **Mandatory Reviews:** Conduct **management and technical reviews, audits, and inspections to verify compliance and performance**

CORRECTIVE ACTIONS

- After an internal or external audit, an organization must perform CAs **to correct the observed deficiencies**.

- The CA (Corrective Action) process can also be used to **address preventive actions, incident reports, and customer complaints**.

- The CA aims at **eliminating potential causes of non-conformity, a defect, or any other adverse event** to prevent their recurrence.

- This process should cover **software products, agreements and software development plans**.

Corrective Action

Action to eliminate the cause of a nonconformity and to prevent recurrence.

Note 1: There can be more than one cause for a nonconformity.

Note 2: Corrective action is taken to prevent recurrence whereas preventive action is taken to prevent occurrence.

ISO 9000 [ISO 15b]

An intentional activity that realigns the performance of the project work with the project management plan.

PMBOK® Guide [PMI 13]

Preventive Action

An intentional activity that ensures the future performance of the project work is aligned with the project management plan.

PMBOK® Guide [PMI 13]

Corrective Actions Process

- The problems encountered when developing systems that include software, or that occur during their operation, can come from defects in the software, in the development process itself, or in the hardware of the system.

- To facilitate the identification of problem sources and apply appropriate CAs, it is desirable that a **centralized system is developed to track issues through to resolution and to determine their root cause.**

A CA process, in a closed loop, may include the following elements:

Inputs:

- Audit report

- Non-compliance issue

- Problem report

Activities:

1. **Register non-conformities** in the organization's issue tracking tool.

2. **Analyze and validate** the problem to ensure resources are not wasted.

3. **Classify and prioritize** the issue/problem based on impact.

4. **Conduct trend analysis** to identify recurring problems and address them proactively.

Problem Resolution Process

Steps to Address the Problem:

1. **Propose a Solution** – Identify and suggest a fix for the problem.

2. **Solve the Problem** – Implement the solution and ensure it does not create new issues.

   - If the problem cannot be resolved within the project, escalate it to the appropriate management level.

3. **Verify the Resolution** – Check that the issue has been fully resolved.

4. **Inform Stakeholders** – Communicate the resolution to all relevant parties.

5. **Archive Documentation** – Store records of the problem and its resolution for future reference.

6. **Update the Issue Tracking Tool** – Ensure the tracking system reflects the latest resolution status.

Outputs:

- **Resolution File** – A documented report of the issue and how it was resolved.

- **Corrected Software Version** – An updated version of the software with the issue fixed.

Problem report

Priority:_____   Project name: _____ Date: _____

Process name: _____   Phase number: ___Raised by: _____

Number of days to answer: _____   Close date: _____

Number of days to fix this problem: _____

Finding: _____

Requirement/Standard impacted:

_____

Immediate solution proposed:

_____

Root cause: _____

Permanent solution proposed:

_____

Acceptance date of permanent solution: _____

Follow-up action (if necessary):

_____

Figure 6.5 Problem report and resolution proposal form.