



## Defect Prevention Techniques Worksheet Template

**Reference:** ISO/IEC 25010:2023, ISO/IEC 25019:2023, ISO/IEC 25002:2024

**Purpose:**

To document and plan preventive quality actions that reduce the occurrence of software defects by addressing their root causes and improving related development and testing processes.

---

### **1** Project and Context Information

Field	Description
<b>Project Name:</b>	Name of the software/system.
<b>Project Phase:</b>	(e.g., Requirements, Design, Development, Testing, Deployment).
<b>Evaluation Reference:</b>	Link to evaluation, audit, or test where the issue or risk was identified.
<b>Prepared By:</b>	Name and role (e.g., QA Engineer, Software Quality Manager).
<b>Date:</b>	
<b>Version:</b>	

---

### **2** Identified Potential Defect Area

This section captures areas where **defects are likely to occur** based on past trends, evaluation findings, or risk analysis.

Field	Description
<b>Module/Component:</b>	Specify affected area (e.g., Login Module, API Layer, Database Schema).
<b>Potential Defect Description:</b>	Describe the defect or failure mode that could occur (e.g., incorrect data handling, delayed response time).



<b>Source of Detection / Insight:</b>	Indicate how this risk was identified (e.g., root cause analysis, code review, lessons learned, audit finding).
<b>ISO 25010 Quality Characteristic Impacted:</b>	Select from: Functional Suitability, Performance Efficiency, Reliability, Security, Maintainability, Usability, etc.
<b>Estimated Severity (1–5):</b>	1 = Low, 5 = Critical.
<b>Estimated Likelihood (1–5):</b>	1 = Rare, 5 = Very Likely.
<b>Risk Score (S × L):</b>	Multiply Severity × Likelihood to prioritize.

### 3 Root Cause Analysis

Before applying prevention techniques, it's vital to **understand the underlying causes** of potential defects.

Field	Description
<b>Suspected Root Cause(s):</b>	Identify likely origins of the issue (e.g., unclear requirements, poor version control, lack of peer reviews).
<b>Detection Method Used:</b>	e.g., 5 Whys, Fishbone Diagram, FMEA (Failure Mode and Effects Analysis).
<b>Supporting Evidence:</b>	Reference data or findings supporting the root cause (e.g., defect logs, test results, post-mortem reports).

### 4 Preventive Action Planning

Plan targeted **preventive actions** that will minimize the likelihood of future defects.

Preventive Technique	Description	Responsible Person	Target Completion Date	Expected Impact on Quality	Status
<b>Static Code Analysis</b>	Use automated			Reduces defect density in code;	



	tools to detect syntax errors, vulnerabilities , and code smells early.			enhances maintainability.	
<b>Peer Reviews / Code Walkthroughs</b>	Conduct systematic peer reviews to identify potential logic or design issues.			Improves reliability and reduces post-release defects.	
<b>Unit Testing Enhancement</b>	Expand unit test coverage for high-risk modules.			Detects low-level logic defects early in development.	
<b>Requirements Validation Workshops</b>	Collaborate with stakeholders to clarify ambiguous requirements.			Reduces functional misinterpretations and rework.	
<b>Static Security Scanning</b>	Implement early security checks in CI/CD pipeline.			Prevents injection and authentication vulnerabilities.	
<b>Developer Training</b>	Conduct targeted sessions on secure coding			Strengthens developer awareness and reduces recurring issues.	



	and quality standards.				
--	------------------------	--	--	--	--

(Add rows as needed based on project scope.)

### 5 Verification and Effectiveness Tracking

After implementation, evaluate whether preventive actions are effective in reducing potential defects.

Field	Description
<b>Verification Method:</b>	e.g., Review updated code metrics, retest modules, trend analysis of defect rate.
<b>Verification Date:</b>	
<b>Verified By:</b>	
<b>Observed Improvement:</b>	Qualitative/quantitative results (e.g., “Defect recurrence rate dropped by 30% over two sprints”).
<b>Effectiveness Rating (1–3):</b>	1 = Ineffective, 2 = Partially Effective, 3 = Fully Effective.
<b>Follow-up Actions:</b>	Additional measures if effectiveness is not satisfactory.

### 6 Lessons Learned and Standardization

Capture insights to embed prevention into organizational knowledge.

Field	Description
<b>Key Lesson Learned:</b>	Summarize what the team learned (e.g., importance of early requirement reviews).
<b>Standard Process Update Required? (Y/N):</b>	Determine if process or policy update is needed.



<b>Document/Procedure to Update:</b>	e.g., Quality Manual, Test Plan Template, Code Review Checklist.
<b>Proposed Preventive Best Practice:</b>	Convert learning into a standardized organizational control or checklist item.
<b>Owner for Update:</b>	
<b>Deadline:</b>	

## 7 Summary Dashboard (Optional)

Visual summary for management reviews or retrospectives.

Metric	Description	Example
<b>Total Preventive Actions Planned</b>	Count of all preventive measures identified	10
<b>% Completed</b>	Completed vs. planned preventive actions	80%
<b>Top 3 Defect Sources</b>	e.g., Requirements, Coding, Testing	Requirements, Coding, Integration
<b>Most Effective Technique</b>	Based on observed outcomes	Peer Reviews
<b>Overall Defect Recurrence Trend</b>	Compare pre- and post-prevention periods	↓ 35% reduction

## 8 Review and Approval

Role	Name	Signature	Date
Quality Manager			
Project Manager			
Technical Lead			



...global validation

QA Lead / Auditor			
-------------------	--	--	--

---

### ✔ Usage Tips

- Integrate this worksheet with **CQI Logs** for ongoing improvement tracking.
- Review it during **Sprint Retrospectives** or **Quality Review Meetings**.
- Use it as **evidence of preventive quality action** during audits or ISO/IEC 25000 compliance assessments.
- Periodically analyze logged data to identify **recurring defect sources** and **update process controls**.