HARDWARE AND NETWORK SERVICING LEVEL-V

Based on December 2024, Curriculum Version - II

Module Title: - Policy Development for Client Support

Module Code:EIS HNS5 03 1221

Nominal duration: 30 Hours

Prepared by: Ministry of Labor and Skills

December 2024

Addis Ababa, Ethiopia

Contents

Acknowledgment ... iii

Acronym ... 1

Introduction to module ... 2

Unit One: Review change requests ... 3

1.1. Receiving hardware and software change requests ... 4

1.2. Gathering and organizing System data ... 10

1.3. Reviewing proposed changes ... 17

1.4. Examining system data ... 19

1.5. Discussing and clarifying changes ... 21

Self-test 1 ... 24

Operation Sheet 1.1 ... 26

Unit two : Modification of IT system ... 27

2.1. Identifying potential solution ... 28

2.2. Developing and documenting solutions ... 31

2.3. Planning implementation and evaluation solutions ... 33

2.4. Documenting recommended solutions ... 33

Self-test 2 ... 36

Operation Sheet 2.1 ... 37

Unit Three : Utilization of modified IT system ... 38

3.1. Preparing and conducting training on changed system ... 39

3.2. Evaluating the training to confirm training effectiveness ... 43

Self-test 2 ... 46

Operation Sheet 3.1 ... 47

Reference 49

Participants of this Module preparation 50

Acknowledgment

Ministry of Labor and Skills wish to extend thanks and appreciation to the many representatives of TVET instructors and respective industry experts who donated their time and expertise to the development of this Teaching, Training and Learning Materials (TTLM)

Acronym

- **EOL** - End-of-Life
- **GPU** - Graphics Processing Unit
- **CPU** - Central Processing Unit
- **I/O** - Input/Output
- **DNS** - Domain Name System
- **IDS** - Intrusion Detection System
- **GDPR** - General Data Protection Regulation
- **HIPAA** - Health Insurance Portability and Accountability Act
- **ELK Stack** - Elasticsearch, Logstash, Kibana
- **SIEM** - Security Information and Event Management
- **SMEs** - Subject Matter Experts
- **QRadar** - IBM Security QRadar (a specific SIEM tool)
- **CAMPER** - Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, Rearrange
- **SWOT** - Strengths, Weaknesses, Opportunities, Threats
- **KPI** - Key Performance Indicator
- **SMART** - Specific, Measurable, Achievable, Relevant, Time-bound

Introduction to module

This module defines the Competence required to Policy Development for Client Support . In this module, introduces Trainees to the foundational concepts and experience in Policy Development for Client Support. This module aims to provide the basic theory, concepts in Determining support areas, Review change request, Modify system according to requested changes and Train on the use of modified system.

Module units

- Reviewing change requests
- Modifying system according to requested changes
- Train on the use of modified system

Learning objectives of the Module

At the end of this session, the students will able to:

- Review change requests
- Modify system according to requested changes
- Train on the use of modified system

Module Instruction

For effective use this module trainees are expected to follow the following module instruction:

1. Read the information written in each unit
2. Accomplish the Self-checks at the end of each unit
3. Perform Operation Sheets which were provided at the end of units
4. Do the “LAP test” giver at the end of each unit and
5. Read the identified reference book for Examples and exercise

Unit One: Review change requests

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:

- Receiving hardware and software change requests
- Gathering and organizing System data
- Reviewing proposed changes
- Examining System data.
- Discussing and clarifying changes

This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to:

- Receive hardware and software change requests
- Gather and organizing System data
- Review proposed changes
- Examine System data.
- Discuss and clarify changes

1.1. Receiving hardware and software change requests

When dealing with hardware or software change requests, it's essential to gather detailed information specific to the technical and operational aspects of the proposed change. Below is a guide on how to document such requests effectively.

Reviewing change requests is a critical process within any Change Management System. It ensures that all proposed changes are properly evaluated, tracked, and implemented with minimal disruption. Below is a structured approach to handling this process effectively.

Key Components to Include in a Change Request Document

1. **Change Request Identification:**
   - ✓ Unique ID or reference number.
   - ✓ Submission date and time.
   - ✓ Name and contact details of the requester.
2. **Description of the Change:**
   - ✓ Clear and concise explanation of the proposed change.
   - ✓ Objectives and expected outcomes.
3. **Reason for Change:**
   - ✓ Justification for the request.
   - ✓ Explanation of current issues or opportunities prompting the change.
4. **Scope of Change:**
   - ✓ Areas or systems affected by the change.
   - ✓ Specific departments or stakeholders involved.
5. **Impact Analysis:**
   - ✓ Potential risks and challenges.
   - ✓ Estimated benefits and improvements.
   - ✓ Dependencies or prerequisites for the change.
6. **Resource Requirements:**
   - ✓ Budget implications.
   - ✓ Human resources or expertise needed.

- ✓ Time and tools required.

7. **Priority Level:**
   - ✓ Classification as high, medium, or low priority based on urgency and impact.
8. **Attachments/Supporting Documents:**
   - ✓ Any additional files or references that provide context (e.g., screenshots, reports, or diagrams).

A change request is a formal proposal to modify hardware, software, or processes within an organization.It can arise from various sources such as users, stakeholders, or system requirements.Common reasons for requests include bug fixes, feature enhancements, compliance needs, or infrastructure upgrades.A change request is a formal proposal to modify a system, process, or configuration to address an issue or improve functionality.

Types of Change Requests

Hardware Change Requests

- Proposals for modifications or updates to physical components of a system.
- **Examples**:
  - ✓ Upgrading server hardware (e.g., adding more memory or storage).
  - ✓ Replacing outdated networking devices (routers, switches).
  - ✓ Adjusting workstation specifications (e.g., replacing processors or upgrading GPUs).
  - ✓ Installing new peripheral devices (e.g., printers or scanners).
- **Common Reasons**:
  - ✓ Performance limitations.
  - ✓ End-of-life (EOL) of existing hardware.
  - ✓ Scalability requirements.
  - ✓ Failures or damage to existing hardware.

Software Change Requests

- **Definition**: Proposals to modify, upgrade, or replace software components.
- **Examples**:
  - ✓ Patching software to fix bugs or vulnerabilities.
  - ✓ Upgrading to a new version of a software application.
  - ✓ Changing system configurations.
  - ✓ Implementing new software features.
  - ✓ Migrating from on-premise solutions to cloud-based applications.

Hardware/Software Change Request Document

A. General Information

1. **Change Request ID:** Unique identifier for the request.
2. **Request Date:** Date and time of submission.
3. **Requester Details:**
   - Name, role, and department.
   - Contact information (e.g., email, phone).

B. Type of Change

1. **Hardware Changes:**
   - Procurement or replacement of devices (e.g., servers, desktops, networking equipment).
   - Configuration adjustments (e.g., memory upgrades, storage allocation).
   - Repairs or maintenance.
2. **Software Changes:**
   - New installations (e.g., enterprise applications, productivity tools).
   - Updates or upgrades to existing software.
   - Configuration changes (e.g., permissions, database settings).
   - Decommissioning outdated software.

C. Description of the Change

- **Summary:** Brief description of the hardware/software change.
- **Details:**
  - ✓ Current state or setup.

- ✓ Proposed state after the change.
- ✓ Technical specifications (if applicable).

D. Business Justification

I. **Purpose of the Change:**

- Enhance performance or efficiency.
- Address existing problems (e.g., system crashes, outdated software).
- Meet compliance requirements.

II. **Expected Benefits:**

- Cost savings.
- Improved functionality or user experience.
- Increased reliability or scalability.

E. Impact Assessment

I. **Scope of Impact:**

- ✓ Systems, applications, or devices affected.
- ✓ Users or departments impacted.

II. **Risk Analysis:**

- ✓ Potential issues (e.g., downtime, compatibility problems).
- ✓ Mitigation strategies.

III. **Dependencies:**

- ✓ Other systems, software, or hardware required for successful implementation.

F. Technical Requirements

1. **Hardware Needs:** Specifications for new devices or components (e.g., CPU, RAM, storage).
2. **Software Needs:**
   - Licensing details.
   - Compatibility requirements.
   - Resource utilization (e.g., disk space, bandwidth).
3. **Support and Maintenance:**
   - Vendor support plans.
   - Internal resource availability.

G. Cost and Resource Estimate

1. **Financial Impact:**
   - Initial costs (e.g., procurement, licenses).
   - Recurring costs (e.g., subscriptions, support contracts).
2. **Resource Allocation:**
   - Human resources required (e.g., IT team, consultants).
   - Estimated time for implementation.

H. Implementation Plan

I. **Timeline:**
   - Proposed start and end dates.
   - Milestones.

II. **Execution Steps:**
   - Detailed plan for executing the change.
   - Testing and validation procedures.

III. **Fallback Plan:**
   - Rollback procedures in case of failure.

Approval Section

I. **Reviewers:**
   - IT manager or team lead.
   - Compliance or risk management officer (if applicable).

II. **Decision:**
   - Approval status (approved, rejected, deferred).
   - Comments or conditions for approval.

Tools for Documenting Change Requests

- **Change Request Templates:**
  - ✓ Use a standardized form for consistency.
- **ITSM Tools:**
  - ✓ Platforms like ServiceNow, Jira, or Spiceworks to streamline documentation and tracking.
- **Documentation Tools:**

✓ Software like Microsoft Word, Excel, or Google Forms to capture and format request details.

Sample Hardware/Software Change Request Template

Change Request Form

Request Details:

- Request ID: ______________________
- Date Submitted: __________________
- Requester Name: __________________
- Contact Information: ______________

Type of Change:

- ☑ Hardware
- ☐ Software
- ☐ Both

Description of Change:

Justification:

Impact Assessment:

- **Affected Systems/Devices:** _________________________
- **Potential Risks:** _________________________________

Technical Requirements:

- **Hardware/Software Specifications:** ________________

Cost and Resource Estimate:

- **Budget Impact:** _________________________________
- **Resource Needs:** ________________________________

Approval Section:

- Reviewed By: ______________________
- Decision: [ ] Approved [ ] Rejected [ ] Deferred

1.2. Gathering and organizing System data

System data refers to the information generated, stored, and processed by the components of an IT system. It includes data related to hardware, software, networks, and user activity, offering insights into system performance, functionality, and health. It plays a crucial role in ensuring that systems perform as intended and provide insights into their behavior, performance, and potential issues.When handling change requests for hardware or software, gathering and organizing relevant system data is crucial for ensuring the change is well-planned, feasible, and has minimal impact on operations.

Key Components of System Data

I. **Hardware Data**:
   - Specifications (e.g., CPU type, memory size, storage capacity).
   - Status and health (e.g., temperature, uptime, power usage).
   - Peripheral devices (e.g., printers, external drives).

II. **Software Data**:
   - Installed software and versions.
   - Licensing and compliance status.
   - Error logs and event histories.

III. **Network Data**:
   - Connectivity status and bandwidth usage.
   - Traffic patterns and routing information.
   - Security data (e.g., firewall logs, intrusion attempts).

IV. **User Activity Data**:
   - Login times and session durations.
   - File access logs and permissions.
   - User error reports and feedback.

Roles of System Data in IT Systems

I. Performance Monitoring
- Tracks system efficiency and utilization rates (e.g., CPU load, memory usage).
- Identifies bottlenecks or underutilized resources.

II. Troubleshooting and Diagnostics
- Provides error logs and events for pinpointing root causes of issues.
- Assists in resolving hardware or software malfunctions.

III. System Optimization
- Highlights areas for improvement (e.g., upgrading outdated components).
- Suggests ways to enhance speed, reliability, or capacity.

IV. Security Management
- Monitors unauthorized access or suspicious activity.
- Tracks vulnerability exploits and strengthens defenses.

V. Compliance and Auditing
- Ensures adherence to organizational or regulatory standards.
- Provides an audit trail for system changes and access events.

VI. Planning and Decision-Making
- Guides infrastructure scaling based on usage trends.
- Supports decisions on system upgrades or replacements.

Purpose of Gathering System Data

- To understand the current state of the affected systems.
- To assess the feasibility, risks, and impact of the proposed change.
- To ensure compatibility with existing infrastructure and processes.

Importance of System Data

- **Monitoring and Maintenance**:
  - ✓ Helps track system health and performance in real-time.
  - ✓ Enables early detection of issues, such as resource bottlenecks or hardware failures.
- **Troubleshooting**:
  - ✓ Provides valuable insights for diagnosing system failures or errors.
  - ✓ Facilitates root cause analysis.
- **Optimization**:
  - ✓ Helps identify inefficiencies and areas for improvement in system performance.
- **Security**:
  - ✓ Detects anomalies or unauthorized activities through log analysis.
  - ✓ Aids in compliance with regulations (e.g., GDPR, HIPAA) through audit trails.
- **Planning and Scalability**:
  - ✓ Informs decisions on future resource needs based on historical trends.

Collection of System Data

- **Manual Methods**:
  - ✓ Collecting configuration files or logs directly from the system.
- **Automated Tools**:
  - ✓ Monitoring tools (e.g., Nagios, Zabbix, SolarWinds).
  - ✓ Log management tools (e.g., Splunk, ELK Stack).
  - ✓ Performance analyzers (e.g., New Relic, Dynatrace).

Sources of System Data

System data is collected from various sources within an IT infrastructure, each offering specific insights into hardware, software, network performance, and user activities. These sources are integral to monitoring, troubleshooting, optimizing, and securing IT systems.

Hardware Sources

Hardware sources refer to data generated by physical components of the system, including servers, workstations, network devices, and peripherals.

- **System BIOS/UEFI Logs**:
  - ✓ Provides information on hardware configuration (e.g., CPU, RAM, storage).
  - ✓ Logs errors or issues detected during system boot.
- **Hardware Monitoring Tools**:
  - ✓ Tools like **Speccy**, **HWMonitor**, or **CPU-Z** provide detailed system specifications, temperature readings, and performance statistics.
  - ✓ Collect real-time performance data, including fan speeds, voltages, and system health.
- **Smart Sensors on Storage Devices**:
  - ✓ **S.M.A.R.T. (Self-Monitoring, Analysis, and Reporting Technology)** data tracks the health of hard drives (e.g., temperature, read/write errors).
  - ✓ Provides early warnings of potential failures.
- **Peripheral Devices**:
  - ✓ Data from devices like printers, scanners, and external drives through software management tools (e.g., **Printer Management Software**, **Device Manager**) can indicate status, errors, and usage metrics.

Software Sources

Software sources provide data related to system applications, operating systems, and installed programs.

- **Operating System Logs**:

- ✓ **Windows Event Viewer** and **Linux Syslog** capture system events, application errors, security logs, and user activity.
- ✓ Include details on system crashes, user login attempts, and errors from installed software.

- **Application Logs**:
  - ✓ Many software programs (e.g., web browsers, database systems, and custom applications) create logs to capture errors, usage statistics, and performance metrics.
  - ✓ Examples: **Apache Access Logs**, **MySQL Error Logs**, or **Windows Application Logs**.
- **Error Reporting Tools**:
  - ✓ Built-in tools like **Windows Reliability Monitor** and third-party programs like **Crashlytics** can gather application crash reports and diagnostic information for troubleshooting.
- **License Management Systems**:
  - ✓ Tools that track the use, status, and validity of software licenses provide data on software usage and compliance.

Network Sources

Network data tracks communication, bandwidth, and security of an organization's network infrastructure.

- **Network Monitoring Tools**:
  - o **Wireshark**, **Nagios**, and **PRTG** collect data on network traffic, packet analysis, and bandwidth usage.
  - o Identify communication issues, security breaches, or performance bottlenecks.
- **Firewalls and Security Logs**:
  - o **Firewall Logs** (from hardware or software firewalls) capture incoming and outgoing network traffic.
  - o Logs monitor blocked access attempts, intrusion attempts, and policy violations.

- **Routers and Switches**:
  - **SNMP (Simple Network Management Protocol)** data can provide performance statistics, error rates, and utilization metrics for routers and switches.
  - **NetFlow** or **sFlow** can offer detailed traffic patterns and bandwidth usage insights.
- **Intrusion Detection Systems (IDS)**:
  - **IDS/IPS Logs** (from tools like **Snort**, **Suricata**) track potential network intrusions or abnormal behavior, such as unauthorized access attempts or malware signatures.

User Activity Sources

User activity data involves logs and reports generated by users interacting with the system, applications, and network.

- **Login/Logout Logs**:
  - Track user authentication attempts, successful logins, logouts, and failed login attempts.
  - Examples: **Windows Security Logs**, **Linux Auth Logs**.
- **File Access Logs**:
  - Captures when users access, modify, or delete files in the system.
  - Systems like **SharePoint** or **Google Drive** log file access for auditing purposes.
- **Application Usage Logs**:
  - **Browser History**, **Application Event Logs**, and **Remote Desktop Session Logs** track user activity with applications and systems.
  - They are essential for understanding user interactions, behaviors, and troubleshooting user-specific issues.
- **Cloud Service Logs**:
  - **AWS CloudWatch** or **Google Cloud Stackdriver** capture logs from cloud infrastructure, providing insights into user interactions with cloud resources, API calls, and service utilization.

External Sources

External sources can provide additional data useful for monitoring system performance and security.

- **Third-Party Monitoring Services**:
  - Cloud-based services like **Datadog**, **New Relic**, or **UptimeRobot** gather external data on website performance, uptime, and resource usage.
  - Track global user traffic and application performance across multiple regions.
- **Vendor-Specific Data**:
  - Data from hardware or software vendors can include firmware updates, configuration settings, and patches.
  - **System Update Logs** (e.g., Windows Update Logs, **yum** on Linux) track changes, upgrades, and software patches.
- **Security Threat Intelligence Feeds**:
  - External feeds from sources like **VirusTotal**, **AlienVault**, or **Ongoing Threat Intelligence** provide data on global security threats, malware, and vulnerabilities that may impact the system.

Performance Monitoring Sources

Performance monitoring tools aggregate data from multiple system components to analyze overall system performance.

- **CPU/Memory Usage Tools**:
  - Tools like **Task Manager** (Windows), **top/htop** (Linux), or **Activity Monitor** (macOS) monitor CPU, RAM, and process utilization in real-time.
  - **Performance Counters** (e.g., in Windows or **sar** in Linux) record performance data over time to track system health and resource demands.
- **System Monitoring Dashboards**:
  - Tools like **Zabbix**, **Grafana**, and **Nagios** aggregate data from multiple systems and generate visual reports to help administrators understand performance trends and potential issues.

1.3. Reviewing proposed changes

- Reviewing proposed changes is the process of evaluating suggested modifications to systems, processes, or projects to ensure they are necessary, feasible, and beneficial.
- This step involves analyzing the purpose, scope, impact, risks, and resource requirements of the proposed changes before deciding to approve, reject, or revise them.
- Proposed changes refer to modifications suggested to improve, fix, or adapt systems, processes, products, or projects.
- These changes aim to address existing issues, meet new requirements, or optimize performance.
- A proposed change is usually formalized through a change request that outlines the details of the modification, its rationale, and its expected outcomes.

Components of Proposed Changes

Description

- A clear explanation of what the change entails, including:
  - ✓ Specific elements to be modified.
  - ✓ Scope and nature of the change (e.g., addition, removal, or alteration).

Justification

- The reason for the change, such as:
  - ✓ Resolving a problem or bug.
  - ✓ Enhancing functionality or performance.
  - ✓ Meeting compliance or regulatory requirements.
  - ✓ Adapting to new business goals or market demands.

Expected Benefits

- The anticipated outcomes, such as:
  - ✓ Improved efficiency.
  - ✓ Reduced costs.
  - ✓ Enhanced user experience or satisfaction.

- ✓ Increased reliability or performance.

Impact Analysis

- An assessment of how the change will affect:
  - ✓ Systems and infrastructure.
  - ✓ Stakeholders and users.
  - ✓ Processes and workflows.

Implementation Plan

- Steps required to execute the change, including:
  - ✓ Timeline and milestones.
  - ✓ Required resources (manpower, budget, tools).
  - ✓ Testing and validation processes.

Risk Assessment

- Potential risks and challenges, such as:
  - ✓ System downtime or disruptions.
  - ✓ Compatibility or integration issues.
  - ✓ Cost overruns or delays.
  - ✓ Mitigation and rollback plans.

Benefits of Managing Proposed Changes Effectively

- Minimizes Risks: Identifies potential issues before implementation.
- Optimizes Resources: Ensures efficient use of time, budget, and manpower.
- Enhances Performance: Results in better systems, workflows, or products.
- Improves Stakeholder Alignment: Encourages collaboration and consensus.
  Supports Strategic Goals: Aligns changes with long-term objectives

Key Objectives of Reviewing Proposed Changes

- Assess Necessity: Determine if the change addresses a real problem or need.
- Evaluate Feasibility: Ensure the change can be implemented within technical, financial, and operational constraints.

- Analyze Impact: Identify potential effects on systems, users, and stakeholders.
- Mitigate Risks: Recognize and plan for possible challenges or negative outcomes.
- Ensure Alignment: Confirm the change supports organizational or project goals

1.4. Examining system data

- Examining system data refers to the process of analyzing and interpreting information generated or stored by a system to assess its performance, diagnose issues, ensure security, or make informed decisions.
- This practice is crucial for understanding system behavior, identifying trends, and maintaining the reliability and efficiency of IT infrastructure or applications

Key Objectives of Examining System Data

- Performance Assessment: Determine how well the system is functioning by analyzing resource utilization, throughput, and response times.
- Problem Diagnosis: Identify errors, bottlenecks, or failures affecting the system.
- Security Analysis: Detect anomalies, unauthorized access attempts, or vulnerabilities.
- Optimization: Use insights from data to improve system efficiency and reliability.
- Decision-Making: Provide evidence-based insights for system upgrades, changes, or capacity planning.

Components of System Data for Examination

- **Configuration Data**
  - ✓ What It Includes: System settings, software configurations, and hardware specifications.
  - ✓ Purpose: Understanding the current setup to ensure compliance with requirements or identify misconfigurations.
- **Operational Data**
  - ✓ What It Includes: Logs, transaction records, and process execution details.
  - ✓ Purpose: Tracking daily operations to ensure smooth functionality and pinpoint failures.

- **Performance Data**
  - ✓ What It Includes: Metrics such as CPU usage, memory utilization, disk I/O, and network latency.
  - ✓ Purpose: Measuring how efficiently the system is running and identifying areas for improvement.
- **Security Data**
  - ✓ What It Includes: Access logs, intrusion detection alerts, and audit trails.
  - ✓ Purpose: Ensuring the system is protected against unauthorized activities or breaches.
- **Historical Data**
  - ✓ What It Includes: Archived logs, backup snapshots, and previous performance reports.
  - ✓ Purpose: Understanding trends over time and planning for future needs.

Tools for Examining System Data

- Log Analysis: Splunk, ELK Stack (Elasticsearch, Logstash, Kibana), or Fluentd.
- Performance Monitoring: Nagios, Zabbix, New Relic, or SolarWinds.
- Security Monitoring: SIEM tools (e.g., QRadar, Splunk Security), firewall logs, or antivirus reports.
- Data Visualization: Grafana, Tableau, or Power BI for presenting insights.

Benefits of Examining System Data

- Improved Performance: Identifying bottlenecks leads to smoother operations.
- Enhanced Security: Detecting unauthorized activities strengthens defenses.
- Proactive Maintenance: Early identification of issues prevents system downtime.
- Informed Decision-Making: Data-driven insights improve planning and resource allocation.
- Compliance Assurance: Analyzing data ensures adherence to regulations and standards.

1.5. Discussing and clarifying changes

- Clarifying changes is the process of providing detailed explanations and resolving uncertainties about proposed modifications to ensure all stakeholders fully understand their purpose, scope, impact, and implementation.
- Discussing changes refers to the process of engaging stakeholders, team members, and decision-makers in conversations about proposed modifications to systems, processes, or projects.
- Discussing and clarifying changes refers to the process of collaboratively reviewing proposed modifications to ensure all stakeholders understand their purpose, scope, impact, and implementation details.
- This step facilitates open communication, addresses uncertainties, resolves conflicts, and refines the change to align with organizational goals and constraints.

Key Aspects of Discussing and Clarifying Changes

- When discussing and clarifying changes, the goal is to ensure that all stakeholders have a shared understanding of the change, its purpose, and its impact.
- This requires clear communication, collaboration, and addressing concerns or uncertainties. Here are the key aspects to focus on during this process:

Clear and Detailed Explanation of the Change

- Purpose: Clearly state why the change is needed. For example, is it to fix a problem, improve performance, meet new requirements, or align with strategic goals?
- Scope: Define what parts of the system, process, or project the change will affect. What is included and what is not?
- Impact: Explain the expected effects on systems, processes, teams, and stakeholders.

Stakeholder Involvement

- **Engage Relevant Stakeholders**: Include everyone affected by or involved in the change, such as:
  - ✓ **Internal teams** (e.g., development, operations, management)

- ✓ **External stakeholders** (e.g., customers, partners, vendors)
- ✓ **Subject Matter Experts (SMEs)**: Bring in technical or domain experts to help clarify complex aspects.

- **Encourage Open Communication**: Create an environment where stakeholders feel comfortable raising questions, concerns, and suggestions.

Addressing Concerns and Questions

- **Provide Clarifications**: Be ready to explain unclear details, such as why certain decisions were made or how specific challenges will be addressed.
- **Risk Identification**: Discuss potential risks or challenges associated with the change and how they will be mitigated.
- **Resolve Misunderstandings**: Take the time to address any misconceptions or fears about the change.

Detailed Implementation Plan

- **Timeline**: Provide an estimated timeline for when the change will be implemented, including key milestones.
- **Resources**: Clarify what resources (e.g., budget, personnel, technology) are required to execute the change.
- **Roles and Responsibilities**: Define who is responsible for each part of the process to ensure accountability.

Adjusting Based on Feedback

- **Listen Actively**: Pay attention to feedback and suggestions during the discussion. What are stakeholders concerned about? What ideas can improve the change?
- **Revise the Plan**: Be open to making adjustments based on valid concerns or better approaches proposed during the discussion.

Building Consensus and Agreement

- **Agreement on Next Steps**: Ensure that all stakeholders are aligned on the implementation plan and the expected outcomes.

- **Documentation**: Document decisions, action items, and any adjustments made during the discussion for future reference.

Clarifying Communication Channels

- **Ongoing Communication**: Establish clear channels for updates and ongoing discussion throughout the implementation phase.
- **Feedback Mechanisms**: Set up mechanisms to gather feedback post-implementation to monitor the change's effectiveness and make further adjustments if necessary.

Managing Change Resistance

- **Anticipate Pushback**: Be proactive in identifying potential areas of resistance and prepare to address them thoughtfully.
- **Empathy and Support**: Acknowledge the challenges that the change might bring to individuals or teams and offer support during the transition.

Documenting and Reporting

- **Formal Documentation**: Record the details of the change, including its objectives, scope, timeline, and the clarifications provided during the discussion.
- **Reporting**: Provide stakeholders with updates, summaries, and progress reports to ensure ongoing transparency and alignment.

Self-test 1

Instruction: -Read all questions which given below and you to answer the correct one.

I. **Choice the correct answers from the given alternatives**

1. What is the primary purpose of reviewing change requests?

A) To approve all changes without questions

B) To ensure changes align with organizational goals and policies

C) To reject changes that seem unnecessary

D) To create new change requests

2. Which of the following is an essential first step when reviewing a change request?

A) Define the budget for the change

B) Identify stakeholders and gather feedback

C) Establish the urgency of the change

D) Understand the purpose and scope of the change

3. When reviewing a change request, what should you check to ensure it complies with organizational standards?

A) The number of stakeholders involved

B) The compatibility with existing systems or processes

C) The estimated cost of implementation

D) The popularity of the request among users

4. What is a key factor to consider when assessing the feasibility of a change request?

A) Whether the change is recommended by external vendors

B) The resources (time, budget, personnel) required for implementation

C) The aesthetic appeal of the change

D) The level of risk associated with rejecting the change

5. Who should be involved in the review of a change request?

A) Only the IT department

B) Relevant stakeholders and subject matter experts (SMEs)

C) Only the senior management team

D) Just the person who submitted the change request

6. How can you manage the risk of a change request causing disruptions to the current system?

A) Implement the change without any testing

B) Conduct thorough testing and pilot implementations

C) Postpone all changes indefinitely

D) Ignore the risks and focus on speed

7. What type of data is most important to include when documenting the review of a change request?

A) Budget breakdowns

B) Detailed list of stakeholders and their roles

C) Clear explanations of the request's purpose, scope, and expected outcomes

D) A list of unrelated projects

8. What should be done if a change request is deemed too costly or impractical?

A) Automatically reject the request

B) Request additional funding without further evaluation

C) Suggest alternatives or revisions to make the change more feasible

D) Ignore the request and focus on other projects

9. When reviewing a change request, what is the importance of stakeholder feedback?

A) It helps to delay the implementation of the change

B) It ensures that all viewpoints and concerns are considered before proceeding

C) It increases the time required for decision-making

D) It determines the final budget for the change

Operation Sheet 1.1

Operation Title: Reviewing Change Requests
Purpose: To Review change request
Conditions or situations for the operations:

- ✓ Safe working area
- ✓ Properly operated tools and equipment
- ✓ Appropriate working cloths fit with the body

Equipment Tools and Materials:

- ✓ Change Management Software

Quality Criteria: Assured performing of all the activities according to the procedures

1. **Initial Screening:**
   - ✓ Verify completeness of the change request form.
   - ✓ Ensure the request aligns with organizational goals.
2. **Impact Assessment:**
   - ✓ Assess potential risks and benefits.
   - ✓ Consider financial, technical, and operational implications.
   - ✓ Evaluate effects on customers, employees, and other stakeholders.
3. **Feasibility Analysis:**
   - ✓ Determine if the change is technically and practically achievable.
   - ✓ Assess resource availability and timeline feasibility.
4. **Consultation with Stakeholders:**
   - ✓ Discuss the proposed change with impacted teams and individuals.
   - ✓ Collect feedback and address concerns.
5. **Approval Process:**
   - ✓ Submit the request to the Change Advisory Board (CAB) or relevant decision-making body.
   - ✓ Use predefined criteria for approval, modification, or rejection.
6. **Documentation of Decisions:**
   - ✓ Record the outcome of the review (approved, rejected, deferred).
   - ✓ Document reasons for the decision and any conditions or modifications required.

Unit two : Modification of IT system

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:
• Identifying potential solution • Developing and documenting solutions • Planning implementation and evaluation solutions • Documenting recommended solutions This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to: • Identify potential solution • Develop and documenting solutions • Plan implementation and evaluation solutions • Document recommended solutions

2.1. Identifying potential solution

- Identify potential solutions refers to the process of recognizing and exploring different approaches or strategies to address a problem or challenge.
- It involves brainstorming, researching, and analyzing various options that could effectively resolve an issue or improve a situation.
- This stage is part of problem-solving or decision-making, where multiple possible solutions are considered before selecting the most suitable one based on criteria such as feasibility, cost, impact, and resources available.
- To identify potential solutions, it's helpful to follow a structured approach.
  - ✓ Define the Problem Clearly: Understand the problem or challenge in detail. Break it down into smaller, manageable parts if needed.
  - ✓ Brainstorm Possible Solutions: Think of a variety of solutions. Don't judge them yet—focus on generating as many ideas as possible, both conventional and creative.
  - ✓ Research Existing Solutions: Look into how others have solved similar problems. Consider existing tools, methods, or strategies that might apply.
  - ✓ Evaluate Feasibility: Assess the practicality of each potential solution. Consider factors like time, cost, required resources, expertise, and potential barriers to implementation.
  - ✓ Analyze Pros and Cons: Compare the strengths and weaknesses of each potential solution. Think about the short-term and long-term impacts, and weigh them against each other.
  - ✓ Consult Stakeholders: If possible, involve others who may be affected by the solution or have expertise in the area. Their insights could help refine or improve the options.
  - ✓ Prioritize Solutions: Based on your evaluation, select the most promising solutions to move forward with. Consider criteria such as urgency, effectiveness, scalability, and cost.

- ✓ Test or Prototype: If feasible, test the solutions on a small scale or develop prototypes to assess their effectiveness before full-scale implementation.
- ✓ Make a Decision: Choose the solution that offers the best balance of feasibility, impact, and resources, and make a plan for its execution.

Tips to Identify Potential Solutions

Define the Problem Clearly

- Write down the problem in simple terms.
- Break it into smaller components if it’s complex.
- Use tools like root cause analysis or 5 Whys to ensure you’re addressing the actual issue.

Involve the Right People

- Collaborate with stakeholders, team members, or subject matter experts who can bring diverse perspectives.
- Encourage open communication and idea-sharing.

Brainstorm Creatively

- Conduct brainstorming sessions where all ideas are welcomed without judgment.
- Use techniques like:
  - ✓ Mind Mapping: Visualize how different ideas connect.
  - ✓ Reverse Thinking: Think about how you might worsen the problem, then reverse those ideas into solutions.
  - ✓ SCAMPER Technique: Modify existing ideas using steps like Substitute, Combine, Adapt, Modify, Put to another use, Eliminate, and Rearrange.

Research and Benchmark

- Investigate how similar problems have been solved in the past.
- Look for case studies, best practices, or industry standards.
- Study competitors or analogous situations for inspiration.

Think Outside the Box

- Challenge assumptions about the problem and its possible solutions.
- Encourage "wild" or unconventional ideas to explore untapped opportunities.

Use Structured Tools

- Employ problem-solving frameworks such as:
  - ✓ SWOT Analysis (Strengths, Weaknesses, Opportunities, Threats).
  - ✓ Cost-Benefit Analysis to assess practicality.
  - ✓ Design Thinking for user-centered innovation.

Prioritize Feasibility

- Consider the resources available (time, money, skills, etc.).
- Focus on solutions that align with the goals and constraints of the project or organization.

Create a Safe Environment

- Encourage a culture where ideas are not criticized immediately.
- Make participants feel comfortable sharing their thoughts openly.

Simulate and Test Ideas

- Prototype or test ideas on a small scale to validate their effectiveness.
- Use feedback from testing to refine the potential solutions.

Stay Flexible

- Be prepared to pivot or combine solutions if new information arises.
- Avoid becoming attached to one idea too early in the process.

2.2. Developing and documenting solutions

- Developing and documenting solutions refers to the process of refining ideas into actionable plans and recording them in a structured format for clarity, communication, and execution.
- Developing solutions involves refining potential ideas and transforming them into actionable plans to address a problem or meet a goal. The goal of this process is to ensure that the solutions are effective, feasible, and aligned with the objectives at hand.
- Documenting solutions is the process of recording the details of a solution in a clear, structured, and accessible format.

Purpose of Documentation

- Clarity: Clearly communicate the problem, the chosen solution, and how it will be implemented.
- Alignment: Ensure all stakeholders are on the same page.
- Accountability: Define roles and responsibilities for implementation.
- Tracking: Provide a reference for progress monitoring and future evaluati

Tips to Developing and documenting solutions

Understand the Problem Thoroughly

- Make sure the solution directly addresses the root cause of the problem.
- Refer to data, feedback, or research to ensure the solution aligns with the identified issue.

Collaborate with Key Stakeholders

- Involve relevant team members or stakeholders early in the process.
- Gather insights, perspectives, and expertise to refine and validate the solution.

Break Down the Solution

- Divide the solution into clear, actionable steps.
- Prioritize tasks and identify dependencies to streamline implementation.
- Ensure the steps are logical, measurable, and easy to follow.

Be Detailed but Concise

- Include enough detail to ensure clarity but avoid overcomplicating the document.
- Use bullet points, tables, or diagrams to simplify complex ideas.

Use Clear and Consistent Language

- Write in plain, straightforward language that can be understood by all stakeholders.
- Avoid jargon unless it's necessary and well-understood by the audience.

Include Visual Aids

- Add charts, timelines, flow diagrams, or illustrations to enhance understanding.
- Use visuals to highlight key parts of the solution, such as workflows or responsibilities.

Focus on Feasibility

- Consider the resources (time, budget, skills, and tools) needed for the solution.
- Address potential challenges and how they'll be mitigated.

Provide Clear Roles and Responsibilities

- Assign specific tasks to individuals or teams.
- Define roles clearly so everyone understands their part in the implementation.

Set Measurable Goals

- Define metrics or Key Performance Indicators (KPIs) to evaluate the success of the solution.
- Use SMART criteria (Specific, Measurable, Achievable, Relevant, Time-bound).

Anticipate Risks

- Document potential risks and challenges associated with the solution.
- Provide contingency plans or strategies for managing those risks.

Use Templates

- Use templates for consistency and structure. Common sections include:
  - ✓ Problem Summary
  - ✓ Solution Overview
  - ✓ Steps for Implementation
  - ✓ Resource Allocation

- ✓ Timeline
- ✓ Risk Management
- ✓ Evaluation Metrics

Review and Revise

- Share the document with stakeholders for review and incorporate their feedback.
- Double-check for accuracy, clarity, and completeness.

Store and Share

- Save the document in a central, easily accessible location.
- Distribute it to all relevant parties and ensure everyone understands the next steps.

2.3. Planning implementation and evaluation solutions

- Implementation and Evaluation of Solutions are two crucial stages in the problem-solving process. These phases ensure that a solution is executed effectively and its success is assessed against predefined criteria.
- Implementation refers to the process of putting the solution into action. This involves planning, organizing, and executing the necessary steps to ensure the solution works as intended.
- Evaluation involves measuring the effectiveness of the solution after it's implemented. The goal is to determine whether the solution met its objectives and whether it delivered the expected outcomes.
- Planning the implementation is about organizing, executing, and overseeing the steps necessary to apply the chosen solution.
- After implementing the solution, the next critical step is to evaluate its effectiveness. This ensures that the solution is achieving the intended outcomes and provides insights for improvement.

2.4. Documenting recommended solutions

- Documenting recommended solutions is an essential step in ensuring that the solution is clear, actionable, and can be effectively communicated to stakeholders.

- It provides a structured way of outlining the rationale, scope, steps, and benefits of a proposed solution, which can be referenced and evaluated over time.

Key Elements of Documenting Recommended Solutions

- **Solution Overview**
  - ✓ **Description**: Provide a brief summary of the solution being recommended. This section should answer the question: What is the solution?
  - ✓ **Example**: "Implementing an AI-driven customer support Chabot to automate basic inquiries and reduce response time."
- **Objective and Purpose**
  - ✓ **Objective**: Clearly state the main objective(s) of the solution. What problem is this solution addressing, and what specific goals are being pursued?
- **Scope of the Solution**
  - ✓ **Inclusions**: Define the scope of the solution. What will be addressed by this solution?
  - ✓ **Exclusions**: Specify what is not covered by this solution to avoid ambiguity.
- **Target Audience**
  - ✓ **End Users**: Identify who will benefit from or use this solution (e.g., customers, internal teams).
- **Implementation Strategy**
  - ✓ Outline the main steps involved in implementing the solution. This section should give an overview of how the solution will be put into action.
- **Timeline**
  - ✓ Provide a high-level timeline for the implementation of the solution. Highlight key milestones and deadlines.
- **Resources Required**
  - ✓ **Human Resources**: Identify the teams or individuals who will be responsible for executing the solution.

- ✓ **Financial Resources**: Estimate the cost involved in implementing the solution, including software, training, and any external consultants.
- ✓ **Material Resources**: Specify any tools, software, or equipment required to implement the solution.

- **Risk Management and Mitigation Plans**
  - ✓ **Identify Risks**: List any potential risks or challenges that could arise during the implementation or operation of the solution.
  - ✓ **Mitigation Strategies**: Propose strategies to minimize or mitigate these risks.
- **Expected Benefits**
  - ✓ **Efficiency Gains**: Explain how the solution will improve efficiency, reduce costs, or improve overall operations.
  - ✓ **Customer Satisfaction**: Highlight how the solution will benefit end-users, such as improving customer experience.
- **Evaluation Metrics**
  - ✓ Define how the success of the solution will be measured. These should be specific, quantifiable indicators that help assess whether the solution is achieving its goals.
- **Conclusion**
  - ✓ Summarize the key points of the recommended solution, reinforcing why it's the best choice.

Self-test 2

Instruction: -Read all questions which given below and you to answer the correct one.

Say true the statement is correct of False the statement is not correct alternatives

1. _______Modifying an IT system often requires careful testing to ensure it does not introduce new issues or vulnerabilities.
2. _______The modification of an IT system always involves a complete overhaul of the existing infrastructure.
3. _______IT system modifications should only be made when there is a critical issue or failure within the system.
4. _______A detailed change management process is essential to ensure smooth modifications of IT systems.
5. _______All modifications to IT systems should be documented to ensure compliance and future reference.
6. _______IT system modifications are typically done by end-users to improve system performance without involving IT professionals

Operation Sheet 2.1

Operation Title: Modifying IT System

Purpose: To Modify IT system

Conditions or situations for the operations:

- ✓ Safe working area
- ✓ Properly operated tools and equipment
- ✓ Appropriate working cloths fit with the body

Equipment Tools and Materials:

- ✓ Network Monitoring Tools

Quality Criteria: Assured performing of all the activities according to the procedures

Step 1. Identify the Need for Change

Step 2. Plan the Change

Step 4. Test the Change

Step 5. Implement the Change

Step 6. Review and Document

Step 7. Close the Change

Unit Three : Utilization of modified IT system

This learning guide is developed to provide you the necessary information regarding the following content coverage and topics:
• Preparing and conducting training on changed system • Evaluating the training to confirm training effectiveness This guide will also assist you to attain the learning outcomes stated in the cover page. Specifically, upon completion of this learning guide, you will be able to: • Prepare and conduct training on changed system • Evaluate the training to confirm training effectiveness

3.1. Preparing and conducting training on changed system

When preparing and conducting training for a newly implemented or changed system, it's crucial to ensure that participants are adequately prepared to adopt and use the system effectively.

Training Needs Analysis

Understand the purpose, process, and methods for conducting a Training Needs Analysis to ensure that training programs are aligned with both organizational goals and client requirements.

Training Needs Analysis is a systematic process used to identify gaps in knowledge, skills, and abilities within an organization or for a specific client. The goal is to determine what kind of training is required to help individuals or groups perform their roles more effectively and efficiently.

Training Needs Analysis:

- Ensuring training is aligned with business objectives.
- Identifying skills gaps in employees or users.
- Prioritizing training resources effectively.
- Reducing training time and costs by focusing only on the essential skills and knowledge.

Key Components of Training Needs Analysis

I. **Organizational Analysis**:

- **Purpose**: Understand the business needs and how training aligns with organizational goals.
- **Activities**:
  - ✓ Review the company's strategic objectives, mission, and vision.
  - ✓ Understand key business operations and how they interact with IT systems.

- ✓ Analyze current and future business requirements to determine areas where training is needed.
- **Outcome**: Identify which organizational goals and processes can be supported or enhanced through training.

II. **Task/Role Analysis**:

- **Purpose**: Identify the specific tasks and responsibilities that need improvement and require training.
- **Activities**:
  - ✓ Analyze current job descriptions and responsibilities of employees.
  - ✓ Assess specific tasks that employees are struggling with.
  - ✓ Determine how the modified IT system impacts job performance and the new skills required.
- **Outcome**: Develop a list of tasks that require specific training intervention.

III. **Person Analysis**:

- **Purpose**: Identify the skills, knowledge, and abilities of individual employees or client groups.
- **Activities**:
  - ✓ Conduct interviews or surveys with employees, team leads, and clients to gather data on current skill levels.
  - ✓ Observe employees in action to assess their competencies.
  - ✓ Perform skills assessments or competency tests.
- **Outcome**: Identify skill gaps for individual employees or client groups and understand their learning styles.

Tips to Preparing and conducting training on changed system

Understand the Audience

- ✓ **Know your learners**: Identify the skill levels, job functions, and familiarity with the system. Customize the content to suit different user groups (e.g., technical staff vs. end users).
- ✓ **Conduct a needs assessment**: Assess what knowledge the audience already has, and focus on the areas that require the most attention.

Develop Clear, Actionable Training Materials

- ✓ **Use simple language**: Avoid jargon and keep instructions clear and concise. If necessary, define technical terms.
- ✓ **Visual aids**: Use screenshots, diagrams, videos, and flowcharts to make the content more accessible and easier to follow.
- ✓ **Create quick reference guides**: Provide cheat sheets, job aids, or checklists to help users on the go.

Choose the Right Training Method

- ✓ **Blended learning**: Combine in-person or live virtual sessions with self-paced materials to accommodate various learning preferences.
- ✓ **Hands-on training**: Allow users to practice in a safe environment, using test systems or sandbox environments where they can interact with the changed system.
- ✓ **Interactive learning**: Include quizzes, group exercises, or scenario-based learning to engage participants actively.

Communicate the Benefits

- ✓ **Emphasize positive outcomes**: Show how the changes will make the user's tasks easier or more efficient, addressing pain points they may have had with the old system.

- ✓ **Motivate early adoption**: Encourage participants to adopt the system by highlighting its benefits, like improved efficiency, time savings, or smoother workflows.

Plan and Organize Training Logistics

- ✓ **Schedule multiple sessions**: Offer several sessions or options (e.g., morning, afternoon, or virtual) to ensure everyone has access to training.
- ✓ **Keep sessions short**: Limit each session to 1-2 hours to maintain attention and retention. Break down complex content into smaller chunks.
- ✓ **Provide opportunities for follow-up**: Allow for refresher training, follow-up Q&A sessions, or one-on-one support as needed.

Facilitate Engagement and Interaction

- ✓ **Encourage questions**: Create an open environment for questions and discussion, either during the training or through dedicated Q&A time.
- ✓ **Interactive demos**: Walk participants through key features of the system, demonstrating how to perform critical tasks.
- ✓ **Peer learning**: If appropriate, encourage experienced users to share their insights or tips with others.

Prepare for Technical Difficulties

- ✓ **Test everything in advance**: Make sure training tools, devices, and the system itself work smoothly before the session starts.
- ✓ **Have a backup plan**: In case of technical difficulties, have a Plan B ready, such as providing alternative content or shifting to another format.

Provide Ongoing Support

- ✓ **Create accessible documentation**: Ensure users can easily access guides, video tutorials, FAQs, and troubleshooting documentation after the training.
- ✓ **Offer continuous help**: Set up a dedicated help desk or support team to answer questions and assist users as they begin using the changed system.

- ✓ **Establish a feedback loop**: Ask for feedback regularly (e.g., through surveys or informal check-ins) to identify ongoing issues and refine training materials accordingly.

Monitor and Measure Success

- ✓ **Evaluate learning outcomes**: Use assessments, quizzes, or feedback forms to evaluate how well participants understand the new system.
- ✓ **Track system adoption**: Monitor usage and adoption rates to assess how quickly and effectively the system is being adopted by users.
- ✓ **Gather feedback**: After training, conduct follow-up surveys to gauge user satisfaction and identify areas where further training or clarification is needed.

Foster a Culture of Continuous Learning

- ✓ **Encourage ongoing learning**: Highlight that training doesn't stop after the first session—continue to offer learning resources, regular updates, and refresher courses.
- ✓ **Create a knowledge-sharing environment**: Encourage users to share their experiences, tips, and challenges with one another, whether through a formal or informal forum.

Be Patient and Flexible

- ✓ **Acknowledge the learning curve**: Understand that adapting to a changed system takes time. Be patient with participants who might struggle initially.
- ✓ **Adapt as needed**: Be ready to adjust the training approach based on feedback or any difficulties participants are facing.

3.2. Evaluating the training to confirm training effectiveness

Evaluating the effectiveness of training is essential to ensure that the objectives were met and that participants are able to apply the knowledge gained.

Key methods for evaluating training to confirm its effectiveness:

1. Set Clear Evaluation Criteria

Before conducting the training, establish clear criteria for success. What specific knowledge or skills should participants acquire by the end of the training? Common evaluation criteria include:

- **Knowledge retention**: Have participants learned the key concepts?
- **Skill development**: Can participants apply the new skills or system functionalities effectively?
- **Behavior change**: Are participants using the new system or processes in their work?
- **Satisfaction**: Are participants satisfied with the training experience?

2. Use the Kirkpatrick Model for Training Evaluation

The **Kirkpatrick Model** is one of the most widely used frameworks for evaluating training effectiveness. It includes four levels of evaluation:

1. **Level 1: Reaction**
   - ✓ **Goal**: Assess participants' immediate reactions to the training.
   - ✓ **Methods**: Use post-training surveys or questionnaires to gather feedback on the training's content, delivery, and overall experience.
   - ✓ **Sample Questions**:
     - How relevant was the training to your job?
     - Did the training meet your expectations?
     - Was the training content easy to understand?
   - ✓ **Purpose**: This helps identify any areas of the training that participants did not find engaging or effective.
2. **Level 2: Learning**
   - ✓ **Goal**: Measure how much knowledge or skill participants have gained from the training.
   - ✓ **Methods**: Use pre- and post-training assessments (e.g., quizzes, tests, skill evaluations) to evaluate learning progress.
   - ✓ **Sample Activities**:

- Pre-training and post-training quizzes to assess knowledge acquisition.
- Practical exercises or simulations to assess hands-on skills.

✓ **Purpose**: This evaluates if the training succeeded in teaching the participants the intended material.

3. **Level 3: Behavior**

✓ **Goal**: Evaluate whether participants are applying what they learned in the workplace.

✓ **Methods**: Observe participant behavior on the job or ask managers/supervisors to assess the application of training.

✓ **Sample Activities**:

- Follow-up surveys or interviews with participants and their managers to see if the skills are being applied.
- Observation of participants performing tasks in the new system or process.

✓ **Purpose**: This level measures if the participants are making the desired changes to their behavior in their work environment.

4. **Level 4: Results**

✓ **Goal**: Assess the overall impact of the training on the organization's objectives.

✓ **Methods**: Measure business or performance outcomes such as productivity, efficiency, or customer satisfaction that are linked to the training.

✓ **Sample Metrics**:

- Time saved due to improved efficiency after system training.
- Reduced error rates or faster processing times due to more effective system use.
- Increased customer satisfaction after implementing a new customer service process.

✓ **Purpose**: This evaluates whether the training has led to tangible benefits and achieved the desired organizational goals.

Self-test 2

Instruction: -Read all questions which given below and you to answer the correct one.

Give short answer for the following questions

1. Why is it important to train users when a modification is made to an IT system?

2. What should be done to ensure the success of a modified IT system after implementation?

3. How can businesses ensure that modified IT systems meet user needs effectively?

4. What role does documentation play in utilizing a modified IT system**?**

Operation Sheet 3.1

Operation Title: Conducting a Training Needs Analysis

Purpose: To Analysis Training Need

Conditions or situations for the operations:

- ✓ Safe working area

Tools:

- ✓ Miro or MURAL
- ✓ Microsoft Teams

Quality Criteria: Assured performing of all the activities according to the procedures

Step 1 Identify Business or Client Objectives:

- Align training objectives with the business goals and needs of the client.

Step 2 Gather Data:

- Collect data that will help determine where the gaps are.

Step 3 Analyze the Data:

- Review and analyze the data gathered to identify training gaps.

Step 4 Define Learning Objectives:

- Develop clear, actionable learning objectives that are tied to the training needs.

Step 5 Develop a Training Plan:

- Create a comprehensive training plan that includes goals, schedules, and resources.

LAP Test 3.1.

Practical Demonstration

Name: ___________________________ **Date:** ____________

Time started: _____________________ **Time finished:** _______________

Instruction: Perform the following tasks

Task 1. Prepare Training Plan.

Task 2. Conduct Training Materials

Task 3. Prepare Post-Training Evaluation Report

Reference

1. "Policy Paradox: The Art of Political Decision Making" by Deborah Stone

2. "Public Policy: An Introduction to the Theory and Practice of Policy Analysis" by Wayne Parsons

3. "The Policy-Making Process" by Charles E. Lindblom

4. "Creating Public Policy: A Practical Guide" by Susan L. M. Ferguson

5. "Handbook of Public Policy Evaluation" edited by David L. Weimer and Aidan R. Vining

Participants of this Module preparation

No	Name	Qualification	Field of Study	Organization/ Institution	Mobile number	E-mail
1)	Abebe Mintefa	MSc	ITM	Ambo TVETC	0929362458	tolabula@gmail.com
2)	Zerihun Abatae	MSc	ITM	Sebata PTC	0911858358	zedoabata2017@gmail.com
3)	Abel G/Egziabher	MSc	IT	MoLS	0911776728	ab.smart99@gmail.com
4)	Mohammed Bekele	BSc	CS	MoLS	0941267468	mbelela@gmail.com