

Information System Implementation

Prepared by Dr. Hesham Mousli

MHA621
Healthcare Informatics

Lecture 7



Intended Objectives

1. Understand **system implementation** and its impact on workflows and efficiency.
2. Learn the **roles and traits** of a successful implementation committee.
3. Explore **workflow analysis, optimization, and testing** strategies.
4. Develop a **training plan** and understand the role of **super users**.
5. Identify **implementation challenges** (e.g., scope creep) and mitigation strategies.

What is System Implementation?

- **Definition:** System implementation refers to the process of introducing new technologies or systems into an organization to improve workflows, enhance patient care, and increase operational efficiency.
- **Analogy:** Like tossing a pebble into a quiet pond, the introduction of a new system creates ripples of change across the organization, affecting workflows, culture, and processes.

Implementation Committee



Forming an Implementation Committee

Chapter 07

- **Purpose:** The implementation committee oversees the planning, execution, and evaluation of the new system.
- **Key Roles:**
 - **Project Leader:** Oversees the entire implementation process.
 - **Clinical Leader:** Ensures the system meets clinical needs.
 - **End Users:** Provide input on workflows and usability.
- **Importance:** Ensures interdisciplinary representation and addresses the needs of all stakeholders.

Characteristics of a Successful Committee

Chapter

07

A. Individual Traits:

- **Open Communication:** Members share ideas and concerns freely.
- **Patience:** Understanding that change takes time.
- **Task-Oriented:** Focused on achieving specific goals.
- **Critical Thinking:** Ability to solve problems creatively.

Characteristics of a Successful Committee

Chapter 07

B. Group Dynamics:

- **Common Goals:** All members work toward the same objectives.
- **Conflict Resolution:** Ability to resolve disagreements constructively.
- **Pride in Accomplishments:** Celebrate milestones and successes.

Planning the Installation



1. *Workflow Analysis and Optimization*

- A. Current Workflow:** Identify existing processes and workflows in the organization.
- B. Gap Analysis:** Compare current workflows with the desired state after system implementation to identify gaps.
- C. Workflow Optimization:** Redesign workflows to improve efficiency and integrate the new system effectively.

***Example: Pharmacy Workflow
Optimization with HIS Implementation***



1. *Workflow Analysis and Optimization*

➤ **Current Workflow (Before HIS):**

- **Medication Orders** → Physicians write paper prescriptions, prone to errors.
- **Prescription Verification** → Pharmacists manually check for interactions, causing delays.
- **Dispensing** → Medications are prepared manually, increasing error risks.

1. Workflow Analysis and Optimization

➤ Gap Analysis:

Process	Current Issues	Optimized with HIS
Medication Orders	Handwritten, prone to errors.	E-prescriptions reduce mistakes.
Verification	Manual interaction checks.	Automated alerts for safety.
Dispensing	Prone to mislabeling/errors.	Barcode scanning ensures accuracy.

1. Workflow Analysis and Optimization

➤ Optimized Workflow (After HIS):

- **E-Prescription & CPOE (Computerized Physician Order Entry)**
→ Eliminates handwriting errors.
- **Automated Drug Interaction Alerts** → Pharmacists receive real-time safety checks.
- **Barcode Medication Dispensing** → Ensures the right drug, dose, and patient accuracy.

Testing the System



2. Testing the System

- **Test Environment:** *A simulated environment where the system is tested before going live.*
- **Types of Testing:**
 - **Unit Testing:** Tests individual components of the system.
 - **Integrated Testing:** Tests how different components work together.
 - **System Testing:** Tests the entire system as a whole.
 - **User Acceptance Testing (UAT):** End-users test the system to ensure it meets their needs.

Hardware Requirements



3. Analyzing Hardware Needs

- **Network Infrastructure:** Ensure the network has sufficient processing power, memory, and bandwidth to support the new system.
- **Workstation Types:** Decide between PCs, thin clients, or mobile devices based on user needs.
- **Device Fairs:** Allow staff to test different hardware options and provide feedback before final decisions are made.



3. Analyzing Hardware Needs

Feature	PCs	Thin Clients
• Performance	High	Depends on server
• Security	Moderate	Very High (no local storage)
• Maintenance	High (individual updates)	Low (centralized updates)
• Cost	Higher (expensive hardware)	Lower (cheaper terminals)
• Flexibility	High	Limited to assigned tasks

3. Analyzing Hardware Needs

➤ Workstation Location Strategies:

- **Point of Care:** Devices located at the bedside or point of care for immediate access.
- **Centralized:** Workstations located at unit stations for shared use.
- **Mobile Devices:** Handheld devices like tablets or smartphones for flexibility and mobility.

Developing Procedures, Training Plan



4. Developing Policies & Procedures

Chapter

07

- **Policies:** Update existing policies to reflect new workflows and system usage.
- **Downtime Procedures:** Develop protocols for system outages, including manual workarounds.
- **User Guides:** Provide clear, step-by-step instructions for using the system.

5. Structuring a Training Approach

➤ Setting Up a Training Environment:

- **Purpose:** Create a safe and effective space for users to learn and practice using the new system.
- **Key Points:**
 1. **Training Database:**
 - Use fictitious or sample data to simulate real-world scenarios.
 - Ensures users can practice without risking errors in live data.

5. Structuring a Training Approach

■ Key Points:

2. Hands-On Practice:

- Allows users to interact directly with the system in a controlled environment.
- Builds confidence and competence through practical experience.

✓ **Why It Matters:** A well-designed training environment reduces anxiety, encourages exploration, and prepares users for real-world use.

5. Structuring a Training Approach

➤ Designing a Training Plan

- **Purpose:** Develop a structured approach to ensure all users are adequately trained.
- **Key Steps:**
 1. **Needs Assessment:**
 - A. Identify the current skill levels of end users.
 - B. Determine specific training needs based on roles and responsibilities.

5. Structuring a Training Approach

■ Key Steps:

2. Schedule:

- Offer flexible training sessions (24/7) to accommodate all shifts and work schedules.
- Ensure no one is excluded due to time constraints.

✓ **Why It Matters:** A tailored training plan ensures that all users receive the right level of support, leading to smoother system adoption.

5. Structuring a Training Approach

➤ Role of Super Users in Training

- **Purpose:** Leverage key individuals to support and enhance the training process.
- **Key Points:**
 1. **Super Users:**
 - Departmental experts who act as the first point of contact for troubleshooting and guidance.
 - Bridge the gap between the IT team and end users.

5. Structuring a Training Approach

- **Key Points:**

- 2. **Additional Training for Super Users:**

- Provide advanced training to ensure they are highly proficient in the new system.
 - Equip them to answer questions, resolve issues, and mentor other staff effectively.

✓ **Why It Matters:** Super users play a critical role in reinforcing training, reducing dependency on IT, and fostering a culture of peer support.

Go-Live Planning



6. Preparing for Go-Live

- **Go-Live Date:** Set a clear target to guide the project timeline and keep the team focused.
- **Key Points:**
 - **Set a Realistic Go-Live Date Early:** Choose a date early in the project that aligns with the system's complexity and organizational readiness to guide timelines and milestones.
 - **Communicate Clearly:** Share the date with all stakeholders to ensure accountability and alignment across the team.

6. Preparing for Go-Live

❑ Key Strategies for Implementation :

1. Staggered Implementation: Roll out the system **in phases** (e.g., by department, location, or function).

➤ **Advantages:**

- Reduces risk by addressing issues in smaller groups.
- Allows time to refine the system before full deployment.

✓ **Best For:** Large or complex organizations needing a gradual, safer approach.

6. Preparing for Go-Live

□ Key Strategies for Implementation :

2. Big-Bang Implementation: Launch the system **organization-wide at once.**

➤ **Advantages:**

- Faster transition to the new system.
- Avoids prolonged dual processes.

✓ **Best For:** Smaller organizations or simpler systems where quick transition is feasible.

6. Preparing for Go-Live

➤ Go-Live Support and Communication:

- **On-Site Support:** Provide 24/7 assistance with adequate staffing to resolve issues quickly.
- **Communication Tools:** Use newsletters, posters, focus groups, and regular updates to keep users informed.
- **Help Desk Role:** Offer 24/7 support and follow up promptly on user concerns.

6. Preparing for Go-Live

➤ Training and User Ownership:

- **Training Challenges:** Avoid rushed training and ensure the system design is finalized before training begins.
- **User Ownership:** Foster user buy-in by involving them in the process and making them feel they own the system.
- **Lack of Buy-In:** Resistance and poor adoption can result from insufficient user involvement.

6. Preparing for Go-Live

➤ Implementation Pitfalls:

- **Scope Creep:** Uncontrolled growth of project scope leading to delays and cost overruns.
- **Over-Customization:** Excessive changes complicate future upgrades; prefer minimal customization.
- **Resource Underestimation:** Insufficient resources cause delays and budget issues.

7. Post-Implementation Evaluation

- **Operational Evaluation:** Assess workflow efficiency, error reduction, and user satisfaction.
- **Technical Evaluation:** Check system stability, reliability, and unresolved technical issues.
- **Financial Evaluation:** Analyze costs vs. benefits and address unexpected expenses.
- **Social/Cultural Evaluation:** Measure impact on user morale and identify cultural barriers to adoption.

8. Continuous Improvement

- **System Maintenance:** Regularly monitor performance, address user feedback, and make updates.
- **Technical Support:** Maintain hardware backups and perform regular system upgrades.
- **Continuous Improvement:** Use evaluations and feedback to drive future enhancements and ensure the system evolves with user needs.

Wrap Up

1. System implementation requires **planning, collaboration, and user involvement.**
2. A strong **implementation committee** ensures effective execution.
3. **Workflow optimization** aligns the system with organizational needs.
4. **Testing** (unit, UAT) resolves issues before go-live.

Wrap Up

5. **Training** with hands-on practice builds user confidence.
6. Choose **go-live strategies** (staggered or big-bang) based on organizational needs.
7. Post-implementation **evaluation** measures success and identifies improvements.
8. **Continuous improvement** ensures long-term system success.



Thank You