Level 1: Elimination – Preventing Errors From Occurring

KEY LEARNINGS: LEVEL 1: ELIMINATION – PREVENTING ERRORS FROM OCCURRING

Overview

- Elimination is the most effective level of service mistake-proofing.
- It's not about catching errors—it's about removing the possibility of errors altogether.
- The goal: Redesign processes so failure simply cannot occur.

What Is Elimination?

- A mistake-proofing strategy that removes error-prone steps, decisions, or options from the process.
- Instead of asking people to "pay more attention," it ensures that no attention is needed for that part of the task.
- Example: Tap-to-pay systems eliminate errors in giving change during cash transactions.

How Elimination Works: The Three Core Methods

1. Removing the Error-Prone Step Entirely

- Target: Steps that frequently cause issues.
- Example: Ride-hailing apps now use GPS auto-detection instead of relying on drivers to enter pickup addresses manually.
- Result: Eliminates miscommunication and wrong pickups.

2. Restricting Choices to Prevent Mistakes

- Target: Tasks with too many options or decisions that can go wrong.
- Example: Digital restaurant menus disable the selection of allergy-triggering ingredients.
- Result: Prevents human error by removing risky choices upfront.

3. Redesigning the Process to Force the Correct Action

- Target: Tasks where the wrong action might still "appear" correct.
- Example: In e-commerce warehouses, packers must scan both the item and the order—if they don't match, the shipment is blocked.
- Result: Physically or digitally blocks incorrect steps from being completed.

Level 1: Elimination – Preventing Errors From Occurring

KEY LEARNINGS: LEVEL 1: ELIMINATION – PREVENTING ERRORS FROM OCCURRING

Real-World Examples of Elimination

- Payroll errors: Automated systems prevent miscalculations.
- Course enrollments: Online portals filter options based on eligibility.
- Utility billing: Smart meters detect vacancy and prevent billing for unused properties.

Key Insight

- Elimination is not about supervision or vigilance—it's about redesigning systems so the mistake never has a chance to happen.
- It transforms unpredictable, error-prone tasks into frictionless, error-proof processes.