**Requirements Management Pain Points (092325)**

|  |  |
| --- | --- |
| **Pain Point Groups** | **Pain Point Group Definitions** |
| Systems are disconnected, forcing manual data transfers | Disconnected tools require frequent manual copying and reconciliation, causing errors, misalignment, and wasted effort. |
| Requirements and test artifacts are stored in incompatible locations and formats | Requirements and test plans live in scattered, incompatible repositories and formats, blocking reuse and slowing validation. |
| End-to-end traceability across the lifecycle is fragmented | Requirements, decisions and downstream artifacts cannot be traced reliably, preventing impact analysis, verification and auditability. |
| Change information is not synchronized across artifacts and teams | Changes are not propagated consistently across artifacts and systems, so revision history, impacts and synchronization are lost. |
| Requirements are captured inconsistently across projects | Project-to-project variability and missing standard processes produce ambiguous, missed or late requirements. |
| Requirements lack sufficient detail and clarity | Inputs delivered to teams often lack the detail and structure needed for design and verification. |
| Stakeholders (including manufacturing and commercial) are engaged too late | Downstream and commercial stakeholders are not involved or their inputs are not captured early, causing rework, misalignment and scope issues. |
| No centralized reusable requirements library exists | There is no single, product-family or standards-based repository to reuse validated requirements, increasing duplicate work and inconsistency. |
| Collaboration and governance practices are inconsistent | Roles, decision rights and collaboration models vary by team, causing delays, confusion and poor cross-functional outcomes. |
| Decisions and customer inputs are not documented in real time | Decisions, customer context and rationale are not captured as they occur, causing loss of tribal knowledge and inconsistent execution. |
| High requirement volatility causes project fragmentation and scope instability | Frequent, uncontrolled requirement changes fragment projects, increase rework, and raise delivery risk. |
| Processes, workflows and tooling are confusing or non‑intuitive | Poorly designed processes and non-intuitive tool workflows reduce productivity, create errors, and increase onboarding friction. |