



Project: **Inventory Manager**

Document: **Risks and Obstacles**

## 1. Incomplete Technical Specification

- Many backlog items are defined only at a conceptual level (e.g., “Feature1”, “Integration1”), without clear business logic or acceptance criteria.
- Lack of detail may lead to misalignment between client expectations and the implemented scope.  
→ **Mitigation:** Conduct backlog grooming and clarify requirements with the client before estimation freeze.

## 2. Integration and Data Flow Uncertainty

- The backlog lists integrations but does not specify external systems (ERP, accounting, logistics tools, or APIs).
- Potential differences in authentication, data structure, or synchronization frequency could impact architecture and delivery timelines.  
→ **Mitigation:** Plan a short *Technical Discovery* phase to map all integration points and data exchange formats.

## 3. Non-Functional Requirements (NFR) Gaps

- Sections like “Security” and “Performance” are mentioned but lack quantitative targets (e.g., response time, load, data retention).
- This makes it difficult to assess infrastructure and scalability needs early on.  
→ **Mitigation:** Define measurable NFRs and include them in the backlog as separate items.

## 4. Environment and DevOps Readiness

- Items such as “Infrastructure” and “Local Dev Environment Setup” indicate early technical setup but do not specify CI/CD pipelines, deployment environments, or cloud provider constraints.  
→ **Mitigation:** Prepare a lightweight *DevOps Readiness Checklist* during the presale phase.

## 5. Security and Compliance Risks

- No details provided on data protection, user roles, or compliance with standards (GDPR, ISO 27001, etc.).
- Security implementation cost can vary greatly depending on regulatory scope.  
→ **Mitigation:** Validate security and compliance scope with stakeholders early.

## 6. Underestimated QA and Maintenance Effort

- Backlog includes “Bug fixing and maintenance” as a single item, which may underestimate the actual QA workload for multi-module systems.  
→ **Mitigation:** Add separate estimation for QA automation, regression testing, and stabilization phase.

## 7. Role Distribution and Resource Constraints

- The backlog lists mainly “Dev” positions, without clear allocation for BA, QA, UX/UI, or DevOps.  
→ **Mitigation:** Adjust effort estimates by adding cross-functional roles to ensure realistic delivery.

## 8. Communication and Decision-Making

- “Meetings with Customer” appear as backlog items — suggesting that project communication itself might require dedicated coordination.  
→ **Mitigation:** Establish a structured communication plan and clarify the RACI model before project start.