Technical Debt Management and Refactoring Strategy

Document ID: IE-TDS-2025.01
Version: 1.0
Effective Date: 2025-11-01
Department: IT & Engineering - R&D

1. Policy Statement

Technical Debt (TD) is an accepted consequence of speed and innovation. However, unchecked TD leads to stagnation, security vulnerabilities, and high maintenance costs. NexaCore actively manages TD to prevent excessive accumulation.

2. Classification of Technical Debt

Classification	Example	Impact/Priority
Critical Debt (P0/P1)	Security vulnerabilities, unscalable architecture components, failing core dependency.	Must be addressed **immediately** (within the current sprint).
Strategic Debt (P2)	Outdated libraries, complex codebase hindering new features, low test coverage.	Scheduled as part of the mandatory **20%** refactoring time in future sprints.
Minor Debt (P3)	Non-standard coding style, poor documentation, minor duplication.	Addressed opportunistically during feature development.

3. Debt Allocation Strategy

- **Dedicated Time:** All engineering teams must dedicate a minimum of **20% of their total sprint capacity** to TD remediation. This time is mandatory and cannot be sacrificed for feature work without CTO approval.
- **Tracking:** TD must be explicitly recorded in the project management backlog (JIRA) with a clear description, impact assessment, and estimated cost to fix.
- **Architecture Review Board (ARB):** The ARB reviews all major design decisions to minimize the creation of new high-impact debt and evaluates the priority of existing

strategic debt quarterly.

4. Debt Amortization

The goal is to amortize the debt over time, ensuring the net amount of new TD created per quarter is less than the amount of existing TD resolved. The health of the codebase is a key metric in the annual performance review of Engineering Managers.