

# Technical Debt (TD) Prioritization Framework (The Quadrant)

Department: R&D (Owned by Head of R&D)  
Objective: Provide a standardized, objective framework for R&D and Product teams to assess and prioritize technical debt alongside new feature development.

## 1. Classification Methodology

Technical Debt (TD) is assessed based on two core axes: **Impact on the Business** and **Effort to Resolve**.

### 1.1 Impact on the Business (Y-Axis)

How severely does this TD affect our ability to ship, maintain SLOs, or attract/retain clients?

- **High (4):** Causes P0/P1 incidents, blocks critical features, or introduces major security risks.
- **Medium (2):** Causes frequent P2/P3 incidents, slows down R&D velocity (slow deploys), or increases cloud costs.
- **Low (1):** Only affects maintainability, documentation, or code readability.

### 1.2 Effort to Resolve (X-Axis)

- **High (4):** Requires full sprints ( weeks) of dedicated engineering time.
- **Medium (2):** Requires full sprints ( weeks).
- **Low (1):** Requires week of work.

## 2. The Prioritization Quadrant

The total score (Impact + Effort) determines the quadrant and the required action.

Score Range	Quadrant	Action / Treatment	Ownership
8 - 16	Tier 1: High Priority (Quick Wins or Major Risks)	<b>Mandatory &amp; Immediate:</b> Must be budgeted for and addressed within the next 2 quarters. Examples: Major security/complianc	R&D Leadership, Product Manager

		e flaws.	
4 - 7	<b>Tier 2: Planned Allocation</b>	<b>Scheduled:</b> Allocate of R&D capacity each sprint for resolving this debt. Examples: High maintenance cost components.	R&D Engineer
2 - 3	<b>Tier 3: Feature-Adjacent Fix</b>	<b>Opportunistic:</b> Only fix when working in the affected codebase for a feature. Do not schedule dedicated time.	R&D Engineer
1	<b>Tier 4: Accept</b>	Document and accept the debt; revisit annually.	Documentation Lead

### 3. Example Debt Prioritization

Technical Debt Item	Impact (I)	Effort (E)	Score (I x E)	Quadrant
Auth microservice runs on a deprecated framework (Node 8).	4 (Major Security Risk)	4 (High Effort)	16	<b>Tier 1</b> (Immediate Refactor)
Inconsistent variable naming convention in reporting module.	1 (Low)	1 (Low)	1	<b>Tier 4</b> (Accept)

Automated integration tests for the API Gateway are slow ( mins).	2 (Medium - Slows velocity)	2 (Medium)	4	<b>Tier 2</b> (Scheduled Allocation)
---	-----------------------------	------------	---	---